Rekik Haile Magicho

Application of SDN Concept in Mobile Backhaul for Traffic Optimization

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 21/11/2014

Thesis supervisor:

Prof. Jukka Manner

Thesis advisor:

Lic.Sc. (Tech.) Esko Räty

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Rekik Haile Magicho

Title: Application of SDN Concept in Mobile Backhaul for Traffic Optimization

Date: 21/11/2014 Language: English Number of pages:9+69

Department of Networking Technology

Professorship: Networking Technology Code: S-38

Supervisor: Prof. Jukka Manner

Advisor: Lic.Sc. (Tech.) Esko Räty

The proliferation of mobile broadband is driven by the evolution of radio technologies in the access network. The Mobile Backhaul (MBH) needs to evolve to avoid being the bottleneck for bandwidth hungry and delay sensitive data traffic. Internet Protocol (IP) based backhaul solutions offer differentiation based on Quality of Service (QoS) and Class of Service (CoS) and flexible bandwidth planning for the ever increasing data traffic. Additionally, Multiprotocol Label Switching (MPLS) based transport protocols support mechanisms for co-existence of different generation technologies. Operator networks benefit from meticulously performed traffic engineering in the backhaul citing that it is very big portion of their network. They also need an efficient network architecture thus SDN is proposed as the new networking paradigm for MBH in this thesis.

Legacy networks -which are often proprietary- mainly consist of purpose built hardware with Application Specific Integrated Circuits (ASICs). ASIC-based hardware have high performance but exhibit low flexibility. These appliances stand in the way of the upgrade flexibility required by the network operators to facilitate speedy innovation and delivery of new services. Software-Defined Networking's (SDN's) decoupled control plane and forwarding plane architecture simplifies the forwarding plane by collecting intelligence in the controller. It offers a standardized interface realizing the desired programmability of the network elements. SDN encourages vendor agnostic standardized protocols to boost interoperability.

The feasibility of SDN in MBH has been studied in this thesis. Moreover the scalability of the centralized controller has been analysed. This thesis is organized in literature review, practical software project and analysis.

Keywords: Software-Defined Networking (SDN), OpenFlow, Traffic Engineering (TE), Mobile Backhaul (MBH)

AALTO-YLIOPISTO
SÄHKOTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Rekik Haile Magicho

Työn nimi: SDN-konseptin soveltaminen liikenteen optimointiin matkapuhelinverkon liitäntäverkossa (mobile backhaul)

Päivämäärä: 21/11/2014 Kieli: Englanti Sivumäärä:9+69

Department of Networking Technology

Professuuri: Networking Technology Koodi: S-38

Valvoja: Prof. Jukka Manner

Ohjaaja: TkT Esko Räty

Radioverkkoteknologioiden kehitys on johtanut liikennemäärän räjähdysmäiseen kasvuun matkapuhelinverkoissa. Matkapuhelinverkkojen liitäntäverkoille (mobile backhaul) on tullut tarve kehittyä, jotta ne eivät jää tietoliikenneverkkojen pullonkaulaksi tietoliikennekapasiteetin ja viiveiden suhteen. IP-pohjaiset ratkaisut tarjoavat joustavia ratkaisuja palvelunlaadun ja palveluluokkien määrittelyyn kasvavalle liikennemäärälle. Operaattorit saavat etua palvelunlaadun määrittelyjen avulla tehdystä huolellisesta liikennevirtojen hallinnasta (traffic engineering). Operaattorit tarvitsevat myös tehokkaan verkkoarkkitehtuurin hallinnalleen - SDN:ää ehdotetaan tässä diplomityössä uudeksi hallintaratkaisuksi matkapuhelinverkkojen liitäntäverkoille.

Perinteiset verkkoteknologiat, jotka saattavat perustua osittain ei-julkisiin, suljettuihin teknologiaratkaisuihin, koostuvat usein tarkoitukseen rakennetuista laitteista ja mikropiireistä (ASIC). Mikropiirit antavat suuren suorityskyvyn, mutta eivät ole kovin joustavia. Mikäli joustavuutta halutaan lisätä matkapuhelinverkoissa uusien innovaatioiden ja palveluiden avulla, perinteisiä verkkoteknologioita on vaikea mukauttaa tähän malliin. SDN:n toisistaan eriytetyt hallintataso ja liikenteen välityksen taso yksinkertaistavat verkon laitteita, koska verkon äly on siirretty keskitetysti hallintatasolle. SDN rohkaisee käyttämään standardoituja, laitevalmistajista riippumattomia avoimia protokollia, jotka mahdollistavat eri valmistajien laitteiden yhteiskäytön.

SDN:n soveltuvuutta matkapuhelinverkkojen liitäntäverkkoihin on tutkittu tässä diplomityössä. Tämän lisäksi keskitetyn hallintaratkaisun skaalautuvuutta näihin verkkoihin on analysoitu. Diplomityö on jaettu kirjallisuuskatsaukseen, käytännön ohjelmistoprojektiin ja itse analyysiin näiden pohjalta.

Avainsanat: Software-Defined Networking (SDN), OpenFlow, Traffic Engineering (TE), Mobile Backhaul (MBH)

Acknowledgements

I would like to thank everyone who supported me during the realization of this work. To Professor Jukka Manner, I would like to express my gratitude for the guidance, encouragement and constructive feedbacks during this thesis work. To my instructor Lic.Sc.(Tech.) Esko Räty, I am deeply grateful for the opportunity you gave me to conduct this work at Tellabs and for the continuous motivation and feedbacks during the practical work and writing of this thesis.

I would like to offer my gratitude to the following Tellabs(now Coriant) employees past and present: M.Sc. (Tech.) Sari Saranka, thank you for the encouragement and inspiration you offered me. M.Sc. (Tech.), Juhamatti Kuusisaari, I appreciate the generous support you provided me through insights and invaluable suggestions at all phases of the project. M.Sc. (Tech.) Juha-Petteri Nieminen, thank you for always making time for answering my questions and for the sharing of experience. M.Sc. (Tech.) Jutta Kemppainen, thank you for being available to help me with INM related issues. M.Sc. Ilpo Närhi, I would like to thank you for training me on using different tools necessary for my work. I would also like to thank everyone in this company for creating a friendly and supportive environment.

My warmest thanks goes to my loving parents and my amazing sister Mebriye for the love, care and encouragement you have bestowed upon me, through thick and thin. Nicolas Malm, I am grateful that you were there throughout this endeavour supporting and encouraging me. I also appreciate the useful remarks on my thesis. I am very thankful for my beloved extended family and wonderful friends for the love, care, fun and support.

Otaniemi, 21.11.2014

Rekik Haile Magicho

Contents

Abbreviations

ACK	Acknowledgement
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
ASIC	Application Specific Integrated Circuit
ATM	Asynchronous Transfer Mode
BC	Bandwidth Constraint
BFD	Bidirectional Forwarding Detection
BGP	Border Gateway Protocol
BSC	Base Station Controller
CAPEX	Capital Expenditure
CDMA	Code Division Multiple Access
CE	Customer Edge
CoS	Class of Service
CSPF	Constraint Shortest Path First
CT	Class Type
Diffserv	Differentiated Services
DS	Differentiated Services
ECN	Explicit Congestion Notification
EDGE	Enhanced Data rates for GSM Evolution
EPC	Evolved Packet Core
EVDO	Enhanced Voice-Data Optimized
FCAPS	Fault Configuration Accounting Performance and Security
FE	Fast Ethernet
FEC	Forwarding Equivalence Class
FIFO	First In, First Out
FQ	Fair Queueing
GE	Gigabit Ethernet
GPRS	General packet radio service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HSPA	High-Speed Packet Access
HSPA+	Evolved High-Speed Packet Access
IETF	Internet Engineering Task Force
INM	Tellabs 8000 Intelligent Network Manager
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol version 4
ISSLL	Integrated Services over Specific Link Layers
LDAP	Lightweight Directory Access Protocol
LDP	Label Distribution Protocol
LSP	Label Switched Path
LSR	Label Switched Router
LTE	Long-Term Evolution

MAM	Maximum Allocation Model
MBB	Mobile Broadband
MBH	Mobile Backhaul
MLPPP	Multilink Link PPP
MME	Mobility Management Entity
MPBGP	Multiprotocol Border Gateway Protocol
MPLS	Multiprotocol Label Switching
MPLS-TP	MPLS-Transport Profile
MTU	Maximum Transmission Unit
NE	Network Element
NFV	Network Function Virtualization
NMS	Network Management System
NTP	Network Time Protocol
OAM	Operations, Administration and Management
ONF	Open Networking Foundation
OPEX	Operational Expenditure
PBB-TE	Packet Bone Bridge TE
PDH	Plesiochronous Digital Hierarchy
PE	Provider Edge
POS	Packet Over SONET/SDH
PSN	Packet Switched Network
PTN	Packet Transport Network
PWE3	Pseudowire Emulation Edge to Edge
QoE	Quality of Experience
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RAN	Radio Access Network
RDM	Russian Dolls Model
RED	Random Early Discard
RNC	Radio Network Controller
RR	Round-Robin
RSVP	Resource reservation protocol
RSVP-TE	Resource Reservation Protocol for Traffic Engineering
SCCA	SDN Congestion Control Application
SDDC	Software Defined Data Center
SDH	Synchronous Digital Hierarchy
SDN	Software-Defined Networking
SDR	Software-defined radio
S-GW	Serving Gateway
SLA	Service-Level Agreement
SONET	Synchronous Optical Networking
TCP	Transmission Control Protocol
TDM	Time-division multiplexing
TDMA	Time Division Multiple Access
TE	Traffic Engineering

ToS	Type of Service
TTL	Time to Live
UMTS	Universal Mobile Telecommunications System
VC	Virtual-circuit
VLAN	Virtual Local Area Network
VMM	Virtual Machine Manager
VoIP	Voice Over IP
VPN	Virtual private Network
VRF	VPN Routing and Forwarding
W-CDMA	Wideband Code Division Multiple Access
WDRR	Weighted-Deficit-Round-Robin
WFQ	Weighted Fair Queueing
WiMAX	Worldwide Interoperability for Microwave Access
WRR	Weighted-Round-Robin
2.5G	Second-and-a-half-generation
2G	Second-generation
3G	Third-generation
4G	Fourth-generation

1 Introduction

Recent years have seen massive increase in cloud based compute and storage services. These rely on flexible and virtualized platforms scaling to the load at any given time. Following the success of these services, intense research into applying these ideas and concepts to networking has emerged. Software-Defined Networking (SDN) resulted from these researches. It aims to address the challenges posed by the escalation of data traffic. Cisco Global Cloud Index forecasts that annual global data center Internet Protocol (IP) traffic will rise to 8.6 zettabytes by the end of 2018 which means it will almost triple from 2013. By the same year about 78% of tasks are forecasted to be processed by cloud data centers [1]. Additionally 5G research envisions connecting tens of billions of devices to the network in what is termed Massive Machine Communication (MMC) [2]. These devices primarily communicate with each other independently of any human interaction. The traffic patterns would therefore differ substantially from the current situation. This change calls for adaptability in the Mobile Backhaul (MBH).

This thesis proposes SDN architecture based approach to MBH Traffic Engineering (TE). SDN centralizes the control plane of the network in an off-device controller that is separated from the data plane devices that do the forwarding based on the remote controller's decision. It makes the devices of the network programmable thus facilitation adaptability which is desired in the MBH. MBH is a large portion of the mobile network that constitutes transport after the cell towers up to the mobile core, therefore, it is very critical for the success of Mobile Broadband (MBB) and future developments of cellular technologies. TE to optimize the performance of the MBH network is necessary considering the cost benefits it brings to the operators and the Quality of Experience (QoE) it creates to end users. Rearranging network topologies to interconnect networking elements more economically depending on time of day, in order to serve the traffic based on this predictable phases of traffic patterns could significantly reduce overall networking cost. [3] In this thesis, this time based traffic pattern is taken into consideration while doing the experiment.

Operators carry the burden when the explosion of services occur with the development of applications on smart devices. However, they see little of the revenue generated, therefore, they need new ways to monetize their assets. Enabling them to do so, will benefit the whole ecosystem by incentivizing operators with sufficient revenue, to invest more in their network. SDN's simplification of the data plane and the use of general purpose programmable devices can have a great impact for revenue generation. This is possible by decreasing Capital Expenditure (CAPEX) with cheaper and flexible data plane elements. Once SDN architecture is set, service provisioning is automated and management is consolidated in a centralized controller reducing the Operational Expenditure (OPEX) in the long term.

1.1 Research Problem

The evolution of the Radio Access Network (RAN) is driving the massive rise in data traffic. The legacy MBH can only cope up with this change through expensive means which are not feasible and profitable for the network operators. The MBH faces the challenge of being the bottleneck if improvements are not made. CAPEX being a very expensive cost of operator's network, how to extract the maximum value of it is one of the set of problems faced. Efficient distribution of traffic among the available resources needs to employ careful TE methodologies. TE in the current MBH however is rather complex due to the co-existence of different transport technologies.

The variation of traffic patterns throughout the day imposes the need in time-varying optimal traffic distribution. The challenge for operators is to create that optimal state with minimum Operations, Administration and Management (OAM) expenditure while satisfying customers. Traditional TE methods lack the necessary flexibility and responsiveness to satisfactorily deal with today's traffic levels. With more and more businesses demanding Service-Level Agreements (SLAs), operators need greater visibility and control of the network to meet the quality guarantees promised. As the network becomes more heterogeneous, its complexity increases many folds to monitor and provision.

SDN has decoupled control and data planes. The intelligence is collected in the controller, which if not researched to ensure the scalability of it, can pose a serious problem. Communications between controller and devices need to be properly protected and authenticated. Therefore, security needs to be an integral part of the SDN's architecture. The centralized controller can be a single point of failure if the necessary security measures are not taken. Strong encryption mechanisms need to be implemented for the communication between the controller and the network devices as well. As devices multiply in number, applying security becomes complex for controller; making scalability a problem in terms of security too. Availability of the controller needs to be addressed due to the dependence of all nodes on the controller. This is an important research problem due to the fact that the controller is the center of every communication for the forwarding plane. Realtime fault recovery needs to be speedy in order to avoid further disruption in the communication. Therefore another challenge is whether the controller manages to solve realtime issues as fast as it needs to. These are some of the crucial challenges being researched in the SDN community in order to facilitate its wide-scale deployment.

1.2 Research and Implementation Plan

Of the above stated research problems, this thesis discusses the scalability issues faced by SDN, the availability of the controller problem, the CAPEX and OPEX problems of network operators, the incremental introduction of SDN in MBH and the realtime fault recovery problems.

To show the feasibility of SDN in MBH, a testbed is prepared that represents the MBH with 6 routers, a traffic generator and a controller. This is a very simplified representation of the MBH, with few nodes, few traffic sources and few routes. OpenFlow protocol which is a communication protocol between controller and forwarding plane provided the programming interface. Tellabs 8000 Intelligent Network Manager (INM) which is a network service management system is used as a controller in this testbed. A traffic generator which emulates the access and core network elements generates traffic through two cell site routers. INM defines paths, one path is for both traffic sources during optimal traffic utilization and the other path is for the less priority traffic to switch to during high utilization in the shared link. The switching occurs after the controller gets reports about high utilization from the shared link and sends OpenFlow messages to the network devices. The purpose of the application is to showcase that SDN can be used for TE in the MBH. TE is very important in the MBH and one aim of TE is to avoid congestion that results in packet delay and loss.

In order to obtain a basis for estimating device reporting load on the controller, number of nodes N = 10,000 to represent real network situation are taken in calculation to show the feasibility based on the amount of control traffic generated. The calculation is done using chosen OpenFlow message sizes. This is done to have an idea of the load of reporting messages sent by devices on the controller. Moreover, a controller feasibility will be estimated by calculating the number of requests a controller can handle per second.

Assuming the traffic towards the controller becomes too much for it to handle, discussion and future research suggestions are given in the areas of distributed controllers and employing a device to off load some computation so that controller does the most important tasks only. The availability of the controller is discussed and suggested that it needs to have redundancy. Moreover, realtime fault recovery is discussed in order to find ways the controller can tackle the challenge of realtime mentioned in the research problem section above. Additionally, how the SDN's architecture can benefit in TE in the MBH is analysed and its introduction in current networking is discussed.

1.3 Outcomes of the Experiment

The outcome of the experiment showed that high utilization was avoided from the shared link and congestion was avoided. The controller managed to switch traffic through one of the sources to pass through another path different from the shared link. The shared link was relieved due to that and congestion did not occur. Consequently, it showcased SDN can become an important element of the TE by switching less priority traffic on less preferred route to avoid congestion as shown by this experiment.

The experiment performed using 6 nodes to showcase SDN's feasibility has its apparent limitation of having only few nodes, only few traffic sources and few routes. That is why a scalability computation based on OpenFlow reply messages towards the controller was later used using N= 10,000 nodes. How many requests can the controller handle from the network devices it manages was calculated. The result was compared with existing controller performance and was shown scalability can be achieved in terms of data plane replies towards the controller. The number of packets per second was also calculated to estimate if controllers can accommodate the generated traffic. Compared with current server capacity, it was shown the controller can handle traffic from thousands of nodes and even left with much capacity to do other tasks. Generally based on the experiment done for this thesis and based on the calculations SDN can be said to be feasible in the MBH.

The outcome exhibited limitation by oscillation in some use case. Oscillation was caused when the shared link exhibits high link utilization and the low priority traffic is switched to the less preferred route but then the shared link is underutilized because of high priority traffic amount that is less than the low threshold. Then as low threshold is experienced the low priority traffic is switched back to the shared link; however, this again causes high utilization needing another round of switching. This goes on until the traffic situation is changed causing oscillation to happen in the network. Oscillation is undesirable in this network because it wastes the resources of the controller by requiring decision to do switching frequently. In the experiment, averaging or sampling was used so that the system avoids reacting to every change in the generated traffic but it was not enough to avoid fluctuation completely from the outcomes. For future continuation of this research, mechanisms to avoid this limitation need to be employed.

The outcome is very crucial as the experiment tries to simulate a real world situation of time based traffic pattern. Traffic amount varies in different links based on time of the day for example, traffic from an area where a shopping mall is, increases during late afternoon hours when many people are in that area and traffic amount from office areas increase during the day during the usual working hours. Time of day based TE avoids over-provisioning and underutilization in the links which can be caused by varying traffic pattern based on time of day therefore this thesis is a good starting point for showcasing of SDN's application in TE leading to future SDN deployment for the use case of TE. Even though the outcomes have shown SDN has use cases in MBH, in real networks the situation is more complex and global optimization algorithms are needed to be studied in order to analyze the different constraints on the network and optimize for a complex network with conflicting interests and several traffic sources. For this algorithm, overall resource usage and delay are important constraints. This is a future research proposal for the continuation of this work.

1.4 Structure of the Thesis

The structure of the thesis consists of extensive literature review, a software application made to address the research problem and analysis of the application and discussion on other challenges of SDN and suggestions for future research. Chapter 2 argues that the motivation for SDN is the inadequacy of legacy networks. It starts by presenting limitations on current hardware centric approaches; next software driven approach is presented in contrast to this. The architecture of SDN is explored and the crucial elements are elaborated.

Chapter 3 has MBH as a central theme. MBH's significance in the Operators' networks and the need for a smart backhaul is explained. The structure of the MBH and the technologies used in it are outlined for the different generations. Some key technologies are expounded upon. Additionally, the challenges faced by the MBH are elaborated and the impacts SDN had on the evolution of MBH is discussed.

In chapter 4, TE is explained and its significance for network operators is discussed. Its impact in the MBH is emphasized and different optimization methods and processes are assessed. Some technologies which were/are used in TE are elaborated. Congestion management is highlighted being the core element of TE and integral part of this thesis and the application illustrated later in chapter 5. Finally, the challenges in TE and the role SDN can play is examined.

Chapter 5 is about SDN based congestion control application made as SDN proof-of-concept. First, the elements that constitute the application are explored. Then, the layout of the test bed is illustrated followed by the elaboration on the implementation algorithm of the software. Additionally, the demo scenarios are laid out with chosen outcomes shown on this paper.

Chapter 6 is analysis and discussion. The analysis includes discussion on the SDN application presented in chapter 5, brief mathematical approach to show the feasibility of SDN using the controller scalability as a basis, discussion on how to address the research problems and solutions as well as suggestions for future research.

2 Software Defined Networking (SDN)

The rapid increase in the number of devices has exposed the limitations of the current approach to network design and operation. Operator network infrastructure lags behind other components of the telecommunication ecosystem in new service adoption. This chapter discusses what the limitations on the current networking approaches are, followed by the advantages of software based approaches. With that it discusses SDN and its architecture and why a standardized interface is crucial. The current de-facto standard interface -OpenFlow- is described to illustrate what a standardized interface of SDN constituted of.

2.1 Constraints of Hardware Driven Approach

Traditional network devices which are hardware driven are built with Application Specific Integrated Circuits (ASICs) for purpose-built hardware. ASIC based purpose-built designs achieve much higher performance than off-the-shelf hardware that are based on general purpose processors. Thus, hardware driven approaches have been providing good quality service. However these are very costly to make and network operators may need to wait a significant amount of time for deployment of new services based on the time frame of the vendors to make upgrades. In addition, expertise is needed to monitor and run the systems. Decoupling software from hardware decreases the constraint a great deal by increasing programmability for the off-the-shelf hardware. [4]

Currently operators buy equipment in large batches, specifying what features/services they want supported. It limits possibilities for experimentation and for upgrade flexibility. SDN is proposed in this thesis as it focuses more on the network as a whole for flexibility and adaptability than raw performance by dedicated hardware. Instead of high raw peak performance locally on one node, SDN changes the metric to be whole infrastructure utilization. A new approach is needed because of traffic growth, speed of usage change and new revenue source needs. SDN avoids having to dimension for worst case scenarios, which if the predictions come true, would be very difficult to achieve. Implementing SDN, there are many ways to leverage software in the network. The architecture we present focuses on optimising overall network utilization. This helps in reducing CAPEX by delaying purchases and allows for better inter-operability through OpenFlow along with reduced costs. In SDN you buy infrastructure upon which you create a network.

In legacy networks a service specific network is bought and competed on price not what you offered. SDN shifts service creation from standardization to operation to some degree. It takes a service and optimization centric view where you scale infrastructure based on need similar to cloud services. Equipment roles become more inter-changeable due to OpenFlow and other standards, resulting in less distinction between routers for role A or role B. In addition, in traditional networking upgrading a forwarding policy demands making changes manually or semi automatically

to the configuration of the devices which is very restrictive when needed to keep up with frequent changes due to the ever increasing mobile data.

Generally, today's networks are comprised of many legacy technologies which are mostly proprietary and very expensive to change. Thus, it is very difficult to deploy network wide services and protocols. Network operators need to deal with the frequent needs of configuration changes manually and semi dynamically that could be cumbersome.

2.2 Software Driven Approach

A Software driven approach in general is a way of automation and virtualization for many areas which would help to create standardized platform and operations. Using software we can reduce the number of hardware used or increase the level of automation and operational efficiency. It brings about new ways to design, build and operate network elements. Fixing software problems can be done remotely and adding modules to the software is easier than a new element to hardware. In terms of cost reduction, the software driven approach has advantages too; buying and maintaining software is cheaper than hardware. Software is potentially more portable and is made to be far more easily upgradable but hardware needs remaking as a whole in order to modify. A Software defined approach also helps in abstracting system details and increases programmability by using software development tools which lead to automation and easier management.

There are much research about software defined approaches. One example is Software Defined Radio (SDR) research. These research may possibly realize a new paradigm on radio technologies and way of managing them. SDR is a radio which part or all of its operating functionality is defined by a software. [5] A single platform could be programmed to do a variety of functionalities by changing the software. The programmability increases and new designs are easier to apply. Analog components are separated from software components that assists in evolving them independently. This is expected to increase performance and to achieve more functionality. [6]

Virtualization is masking of resources by software to create one or more virtual versions of a resource for the purpose of resource optimization. Virtualization can be done on different levels for example at storage, operating system (level), network and server levels. A hypervisor or a Virtual Machine Manager (VMM) is a software that allocates resources to each virtual machines (VMs) that run on different instances of a device. Software Defined Data Center (SDDC) is another research area when in technology architecture mechanism for achieving IT as a service by using software which is guided by policies. It extends virtualization in a way that functionalities are virtualized into a software service decoupled from hardware. In this paradigm there are different domains; software defined storage, security, networking and computing.

2.3 SDN Architecture Motivation

Network management systems (NMSs) have been alleviating the cumbersome task of managing the network by allowing the network administrators to supervise network elements with software or hardware tools and contributing greatly to TE optimization. NMS could be considered as the basis for software defined paradigm by keeping track of network resources remotely. One example NMS -INM- is summarized in Section 5.1.1. SDN is a paradigm shift towards simplified networking by abstraction of the forwarding plane to enable improved service structure. This creates a new standard and a new infrastructure. It is an improvement on top of the NMS and potentially evolves the NMS because the Fault, Configuration, Accounting, Performance and Security (FCAPS) data in SDN architecture could be retrieved from a single controller rather than from several network devices.

The motivation behind SDN architecture is the inadequacy of the traditional network architecture to support the growing needs of users in the face of the proliferation of mobile broadband. Some of the driving forces behind this escalation of data are the increasing number of mobile devices, server virtualization and cloud services. In the SDN architecture, the control and data planes are decoupled. Network intelligence or the control logic is centralized in the control plane and resulting in the forwarding plane being much less sophisticated (simple packet forwarding devices) when compared to traditional networks. The logically centralized control plane is more efficient than the distributed legacy systems. Moreover, controller based applications do not need to depend on the network infrastructure as the routing decisions are made in the control plane using software. They perform specific functions through the controller using the network information it has. This increases the scalability and the control an operator has over the network. The programmability and automation that result from the control could speed up the time to market of services. The Open Networking Foundation (ONF) is a non-profit industry consortium that is founded to improve SDN and standardize critical elements of the SDN architecture. [7]

As mentioned in [7], current protocols are designed in an independent manner and in order to make any changes, we need to deal with the complexity that arises from the independent workings of the protocols and the interruptions that could arise due to any change. One trend nowadays and seemingly for the future is to use VMs that will hold the applications in a distributed manner. The VMs also exchange traffic flows possibly causing the change in destination of flows over time which in turn poses challenges for traditional networks where destinations are known before hand. This dynamic traffic pattern makes the network difficult to scale and as the network grows manual configuration is not feasible scaling-wise. Independently designed protocols as in the case of traditional networks are distributed and make local decisions. Therefore operators do not have full control because it it very difficult

to predict the overall effect on the network based on local changes. Although at times they can tweak some parameters in the protocols of traditional networks in order to serve some purpose, with SDN the operators have more control to defining the path they want the packets to take much easily.

2.4 SDN Architecture

As mentioned earlier, SDN is a network architecture that creates a logically centralized control plane separate from the forwarding plane in contrast to the traditional networks where the network devices have control and data plane in the same device. Network intelligence is centralized in the in SDN controller which makes the network devices and the infrastructure layer much simpler. The infrastructure is abstracted and can be virtualized. SDN provides a programming interface to define the network with software. The standardized interface is an interface by which the controller communicates to the forwarding plane. OpenFlow is one example of a standardized interface.

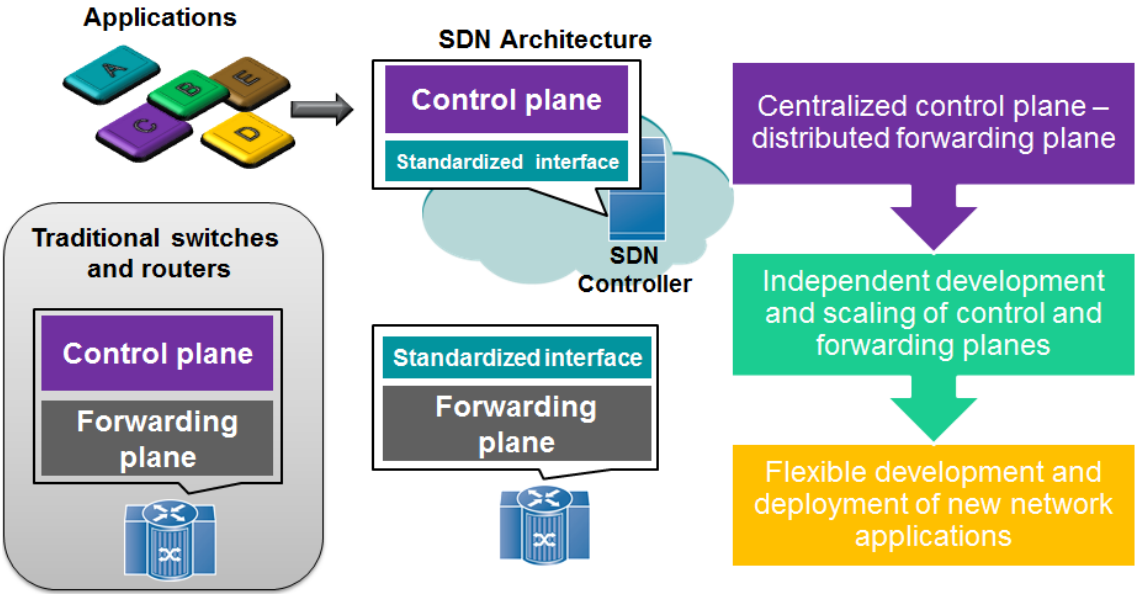

Figure 1: SDN network architecture [8].

The control plane functionality is logically centralized into an SDN controller. Applications are written assuming the network is a single system. The intelligent control plane depicted in the above picture maintains a global view of the underlying infrastructure. This makes it easier for network operators and managers to control the network by only working on a centralized intelligent SDN controller with a standardized interface. This in turn gives the freedom to develop programs vendor-independently. Therefore, applications are developed faster for SDN-like infrastructure than the traditional one as software is faster to program than hardware to design while being cheaper and more interoperable.

Open Application Programming Interfaces (APIs) are being researched on by ONF. They assist in bringing about multi-vendor management which results in an on-demand resource allocation and a secure fully virtualized networking. Thus, using open APIs between the SDN control and application layers, applications can operate on abstractions of the network without needing to be aware of the network but only its capabilities which results in optimization of computing, storage and network resources. [7] One critical element the ONF standardizes is the OpenFlow protocol which is a standardized interface that enables communication between the decoupled control and data planes in SDN. Using SDN with OpenFlow brings about considerable advantages such as centralized management that leads to rapid innovation, increased security and programmability. These bring about better end-user experience through applications that know about the network state and adjust to suit the user. The centralized control also helps SDN to adapt to dynamic user needs and applications to adjust to the network behaviour in order to better the user experience. [7]

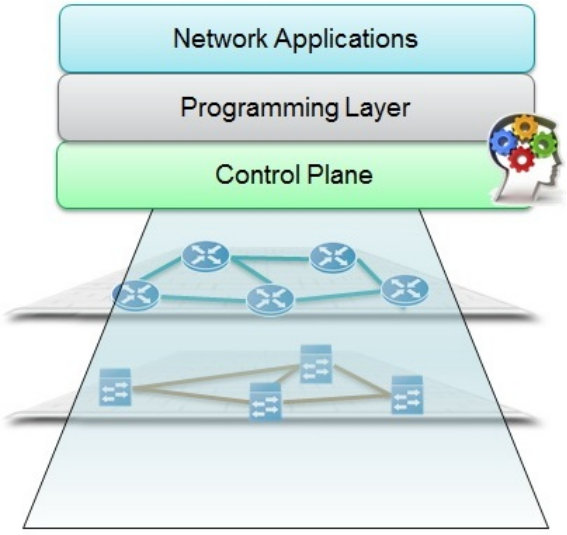

Figure 2: SDN collects the intelligence in the control plane: [8].

In traditional networks without SDN, the intelligence lies in each layer. Figure 2 above depicts that SDN collects the intelligence to a centralized location: the control plane. It also shows SDN has APIs for a programming layer on top of the control plane -northboundand APIs- for the communication of the controller and the services and applications which run over the network. Thus, applications can be developed easily which is an essential feature of SDN that speeds service innovation. The traditional protocols do decision in a distributed manner; however if they can be in the network applications layer with SDN architecture they will have a global view for the network giving them a better understanding of it. In addition, the applications are interfacing to a programming layer and programmability is a powerful feature of SDN. This open network application development brings about new application ecosystem where applications are developed fast and can be used as a differentiator against competitors.

2.5 OpenFlow for SDN

An API for SDN need to be very powerful as it has to support the communication between the two planes. It also needs to keep standards among vendors to avoid proprietary protocols in order to prevent problems the traditional networks have OpenFlow is one such protocol. If it is used, it is set on both the control and the network infrastructure interfaces and gives access to programming of the network devices on per-flow basis to enable SDN networks. ONF standardizes the OpenFlow protocol. ONF states that OpenFlow-based SDNs can be deployed on existing networks and can co-exist with the traditional forwarding and suggests that this makes it easier for the progressive introduction of it to enterprises and carriers. The Open Networking Foundation is chartered to standardize OpenFlow. [7]

The controller can control OpenFlow-enabled network devices from multiple vendors. The complex current network can be simplified by automation that can be provided by OpenFlow-SDN combination. Network security and reliability increases as an OpenFlow based SDN architecture does not need to configure each device individually whenever there is a change in the network reducing the possibility of network failure. Moreover, the global view the controllers have makes security policies easier to apply.

2.5.1 OpenFlow Switch

There are OpenFlow only and OpenFlow hybrid switches. The OpenFlow only switches support only OpenFlow pipeline processing whereby the only function that they provide is to forward packets. Whereas, the hybrid ones can also do the traditional Ethernet switching operations such as L2 switching, L3 routing and Virtual Local Area Network (VLAN) isolation. [9] A hybrid switch is a good approach to the interoperability of SDN during the transition to SDN. Most commercial switches available are hybrid switches at the moment. Examples of currently available commercial OpenFlow enabled switches include NetIron 2000 series and CER 2000 from Brocade, EX9200 programmable switch from Juniper and RackSwitch G8264 and G8264T from IBM. There are software based openFlow switches for testing and developing OpenFlow- based network applications. Some examples are Open vSwitch, Indigo and LINC. Existing controller implementations include NOX, POX, NodeFlow, Floodlight and OpenDaylight. [10]

In OpenFlow there are two important elements: the OpenFlow switch and the controller. Unlike traditional switches, the forwarding and high level routing decision are separated. The controller makes high level routing decisions used to manage the switch which only performs simple forwarding. The communication between the switches and the controller over the secure channel is done over Transmission Control Protocol (TCP) connection that can possibly be encrypted using Transport Layer Security (TLS). TLS encrypts using mutually authenticated certificate signed by a private key that corresponds to a public key trusted by both parties. Controller-to-switch, asynchronous, and symmetric message types are supported by the OpenFlow

protocol. Reliable delivery of message is offered by OpenFlow protocol however, it might not provide acknowledgement or assurance of ordered message delivery. The controller-to-switch messages are from the controller to monitor the state of the switch. Asynchronous messages are sent from the switch to report the state and changes in the switch. Symmetric messages are from both parties without being requested. [9]

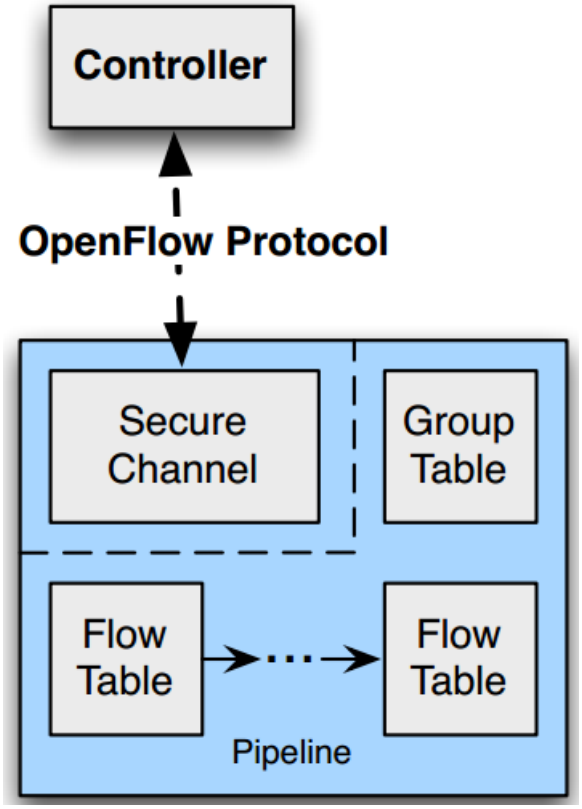

Figure 3: OpenFlow Switch. [9]

Switches must process every message received from a controller in full; otherwise they need to send back an error message. When OpenFlow channel is being set up, the controller inquires features information from the switch to know its capabilities. [9] An OpenFlow switch has one or more OpenFlow tables and a group table. OpenFlow enables data plane abstraction based on flow tables. Each flow table consists of flow entries. A flow entry is depicted in Table 2.

Table 2: Main components of a flow entry. [9]

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

Match fields are set of information to match against packets and priority is a field to indicate the precedence of the flow entry. Match fields have the ingress port,

packet headers and optional metadata which is information passed from previous table. In the flow table, the combination of match fields and priority is used to identify a particular flow entry. Counters are used to track packets and update when matching happens. Instructions are for modifying the action set. Therefore when matching occurs the counters are updated and instructions are executed. The result is changed packet, action set and/or pipeline processing. Timeouts are maximum amount of idle time before a flow is expired and cookie is a value chosen by the controller to filter flows in different ways.

Matching starts at the first flow table and continues to the next table if a match is not found. The tables are numbered sequentially starting from 0. Flow entries send a packet only to higher numbered flow tables. Flow entry that matches the packet's match fields is selected when it has the highest priority among all the matches. When there is a matching flow entry with the packet's match fields, the instructions for that entry are executed and the packet is often forwarded; the counters for that flow entry are updated as well. Instructions execute in order to cause changes to the packet and they contain action set and/or modify pipeline processing. The action set contains actions such as packet forwarding and modification. If the matching flow entry does not send packets to any other tables, pipeline processing halt at this table and the packet is processed with its associated action set. OpenFlow pipeline processing is the mechanism by which packets interact with the OpenFlow tables.

An OpenFlow switch must support needed instructions such as Write-Actions and Goto-Table but it may include optional instructions as well. The Goto-Table instruction is not in the last table of the pipeline at which point the table pipeline processing stops and some actions are performed on the packet. If the instructions associated with the flow entry can not be executed by the switch it rejects the flow and returns unsupported flow error. If there is no matching flow entry table-miss flow entry may drop it, pass it to another table or the packet is sent to the controller. The controller decides whether to drop the packet or add a new entry that helps in forwarding the packet. Table-miss flow entry wildcards all matches and its priority is 0 and it has to be supported by every flow table. The addition, deletion and update of flow entries is done by the controller using the OpenFlow protocol. Flows can be removed by the controller or based on what is set in the flow entries when timeouts are exceeded. [9]

Group tables consist of group entries. Each group entry has group identifier to uniquely identify the group and a group type to determine the group's semantics. Moreover, if packets are processed by a group, the counters in the group entry update. Another component of the group entry is an ordered list of action buckets. They contain a set of actions to execute along with associated parameters.

There are different group types and they can be categorized into "required" and "optional" based on which types a switch needs to support. The required group types are "all" and "indirect". The "all" type means execute all buckets in the

Table 3: Main components of a group entry in the group table. [9]

Group Identifier	Group Type	Counters	Action Buckets

group which is used for multicast or broadcast forwarding. One packet is processed for each bucket of the group. The "indirect" type executes one defined bucket in this group. This group supports only a single bucket. This allows multiple flow entries or groups to point to a common group identifier for more efficient and speedy convergence. The optional types are "select" and "fast failover". The "select" type executes one bucket in the group. Packets are processed by a single bucket selected by a switch calculated algorithm that is non-OpenFlow. The "failover" type executes the first live bucket. This type enables the switch to change forwarding without consulting with the controller. The group defines the order of the buckets and the first packet associated with a live port is executed. [9]

OpenFlow messages start with an 8 bytes OpenFlow header. The header is comprised of the version number of the OpenFlow protocol used. Type represents the type of the message. The total length of the message including the header is represented by the length field. The transaction id is the id associated with this packet.

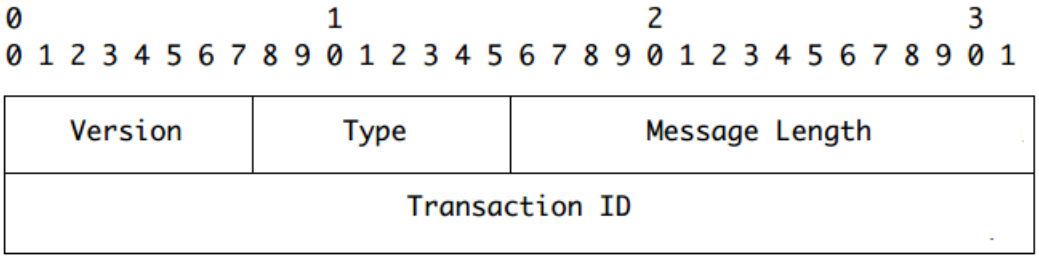

Figure 4: OpenFlow Message Header.

2.5.2 OpenFlow Ports

OpenFlow ports are network interfaces between the pipeline processing and the rest of the network. The ingress ports receives the packets, and depending on the action it could be sent out on the output port. There are three types of ports in the OpenFlow switch these are physical, logical and reserved ports. Physical ports are ports that correspond to a physical interface or virtualized part of a hardware interface. Logical ports are defined by the switch and could correspond to multiple physical ports. Packets linked with logical ports have a tunnel-ID in their metadata and when sent to the controller their logical address and the corresponding physical address need to be reported. The reserved ports are defined by the ONF specification for OpenFlow and they are the ports that do the generic forwarding actions such as forwarding packets to other specified ports or to the controller and, flooding to standard ports. There are required and optional reserved ports and ports that can

be used as an input, output or both. [9]

The required reserved ports are ALL, CONTROLLER, TABLE, IN_PORT and ANY ports. The optional reserved ports are LOCAL, NORMAL and FLOOD ports. ALL represents all ports used for forwarding of a specific packet. It is used as an output port. CONTROLLER port represents the control channel with the OpenFlow controller. It can be an ingress or an output port. TABLE represents the start of the OpenFlow pipeline. IN_PORT represents the packet ingress port and it can be used only as an output port sending the packet out through its ingress port. ANY can not be used as an ingress port nor as an output port. It is a special value used when no port is specified. LOCAL represents the switch's internal organization, the local networking stack and the management stack. It can be used as an ingress port or as an output port. It makes remote elements interact with the switch through OpenFlow via the OpenFlow network rather than a separate control network. NORMAL represents the traditional pipeline if the switch is non-OpenFlow. It can be used only as an output port. FLOOD represents using the normal pipeline of the switch for flooding and can be used only as an output port. The NORMAL and FLOOD ports may only be included in the OpenFlow-hybrid switches. [9]

2.6 Chapter Summary

SDN has a logically centralized controller and a distributed forwarding plane contrary to traditional networking architectures, where the control and forwarding planes are within the same device. SDN aims to address the limitations of the current approach to building operator networks by increasing flexibility and interoperability. The inadequacy of the traditional network architecture, as the data traffic increases dramatically, is the driving force behind SDN. Operators need a way for fast service innovation and delivery and the future towards standardised programmable networks is desirable. The development of the communication protocols between the controller and forwarding devices is crucial element in the realization of SDN. OpenFlow is the de-facto protocol for SDN currently. It is beneficiary in the interoperability of SDN with legacy networking because OpenFlow hybrid switches are able to switch between traditional switching and OpenFlow pipeline processing.

3 Mobile Backhaul (MBH)

Each successive generation of cellular technology brings promise of ever greater service for end users. Achieving these headline performance figures is however not possible without a corresponding evolution in the networks connecting the RAN to the core. This chapter begins by describing the purpose and structure of a typical MBH. Current state-of-the-art key technologies employed are discussed next. Finally this chapter analyses the challenges faced by current MBH and the implications of the move towards all-IP.

3.1 Overview of MBH

MBH is part of the network that carries traffic to the network core in mobile communication; it starts the connection after the cell tower and connects the air interface to the Base Station Controller/ Radio Network Controller (BSC/RNC) and the mobile core. The type of backhaul is the result of a a combination of requirements such as the technology used in the RAN, geographical factors, the bandwidth requirements, transport mechanisms and regulations [11]. The backhaul therefore may consist of different connection types such as microwave antennas, fiber or it could be realised as a combination of both. Radio access networks are advancing to higher speeds, coverage and capacity to accommodate increasing user needs for bandwidth-heavy services. The backhaul solution needs to evolve accordingly because the legacy backhaul cannot provide this high-capacity, high-quality service without improvements currently. [12].

Table 4: Radio access technologies and their transport interfaces [13]

Technology	**Data rates**	**BS/ Node B I/F**
GPRS/EDGE(2.5G)	40Kbps/14Kbps 130Kbps/130Kbps	BS => Voice & Data: A-bis over E1/SDH
W-CDMA(3G UMTS)	384Kbps/384Kbps	Node B => Voice & Data: ATM over E1/SDH
. HSPA(3.5G)	14.4Mbps/384Kbps 14.4Mbps/5.72Mbps	Node B => Voice: ATM over E1/SDH Data: IP over MLPPP/E1 or FE
HSPA+(almost 4G)	28Mbps/11Mbps 42Mbps/11Mbps	Node B => Voice & Data: IP over GE or FE
LTE(4G)	138Mbps/37Mbps	eNode B => Voice & Data: IP over GE or FE

Long-Term Evolution (LTE) and High-Speed Packet Access (HSPA) are IP based, unlike Asynchronous Transfer Mode (ATM) based 3G and E1 based 2G access networks. The future is LTE however legacy backhauls are Time-division

multiplexing (TDM) based using E1/T1 connectivity. These are not feasible for the ever increasing data traffic. LTE demands more capacity as well as very as low latency which some sources state less than 10ms. As a consequence, for a full packet transport, it is recommended that the aggregation must be close to the access networks and have fewer tree and branch hops. [13] Operators are trying their best to meet the fast growing demand for mobile broadband that increases data traffic greatly and they need to invest in creating those capabilities needed for mobile broadband. This increases operators CAPEX and OPEX. Therefore, implementing a smart backhaul solution that meets the requirements needed by users and operators is very important in order to handle the ever increasing mobile traffic in a cost effective manner. [14]

Heterogeneous networks or hetnets are networks that incorporate 2G, 3G and LTE networks. A good backhaul solution that can support all kinds of traffic in the hetnets is needed in order to assure efficient access to users. There are different types of backhauls such as TDM, Ethernet and Multiprotocol Label Switching (MPLS) backhauls. For LTE Carrier Ethernet backhaul is preferred due to high-speed data and flatter core network which is desirable as the traffic is more dynamically distributed in the backhaul network in addition bringing simplicity in deployment. Carrier Ethernet extends Ethernet extensively and adds strong fault tolerance and OAM capabilities to use it in the wide area. Some features of smart backhaul solutions are content awareness in order to orchestrate traffic flow optimization, integrated security technologies to have security gateways cost effectively, disguising hetnets in order for them to appear homogeneous and less complex, enabling any-to-any connectivity by incorporating relevant technologies and having network management mechanism to sustain operational efficiency. [14]. Therefore a good backhaul needs to have holistic understanding of the mobile network.

Operators try to create good QoE by deploying more advanced technologies, gaining access to new spectrum, densifying the macro layer and deploying small cells among other solutions. Densifying the macro layer is splitting the macrocell or adding small cells at some areas. [15] Smaller cells are better for serving bandwidth-intensive data services because of short distances to the hot spot. Small cells can also extend macro cells in order to extend coverage. The small cells are low power cells and cover smaller areas and they can be used for smaller distances to unburden the high power macro cell. This helps the macrocell to serve farther areas giving a chance to take the traffic to more users without adding macrocells. Moreover small cells have lower hardware cost and reduced overhead for power amplification and cooling [16]. Macrocells serve a much larger area and high-speed users in addition to coordinating microcells, picocells and other smaller cells. A good backhaul solution need to create uniform QoE whether users are covered with macrocells or small cells. The use of more microcells results in more nodes/more base-stations thus more connections. SDN can prove useful in managing the growing number of nodes with a central controller.

3.2 Structure of MBH

There are different kinds of routing technologies within the MBH. The technologies can be IP based as Ethernet and MPLS or a mixture of other technologies such as ATM and Frame Relay [17].

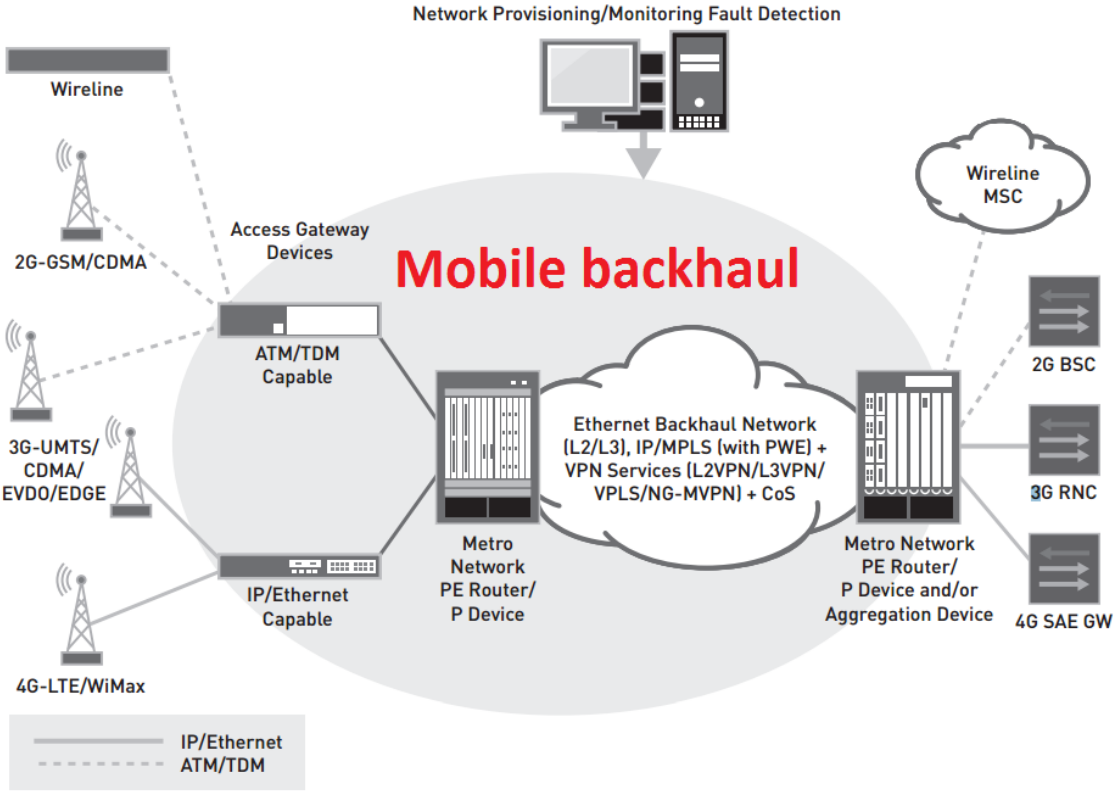

Figure 5: Structure of MBH network [17].

Figure 5 shows an overview of an example Ethernet backhaul network. As shown in the figure, MBH spans from the last domain of the cell site through the aggregation of access and metro domains. The cell sites consists of devices that use technologies of different generations. The sites access the Metro Ethernet network through different access gateways that can support connectivity to a variety of technologies. It is depicted that MPLS is used as a transport mechanism with pseudowire for services that use the backhaul transport and can pass multiple technologies over the same physical link transparently. The aggregation of the metro as shown in the figure aggregates all incoming connections before sending them off to the mobile core. [17].

As mentioned earlier, to come up with the backhaul solution different factors are taken into account. One of the factors is the radio technologies used. Table 1 below shows the access and backhaul solutions applied when we use different technologies. [17]

Table 5: Mobile Technologies and base Station Support [17]

Mobile Network Generation	**Technology**	**Base Station Interface**	**Base Station Support**	**Backhaul Network Support**
2/2.5G	GSM/ GPRS/ TDMA/ CDMA	Abis	Channelized TDM	PDH/SDH
3G [Rel 99]	UMTS	lub	ATM	ATM
3G/4G	EVDO, UMTS[Rel5], WiMAX, LTE	lub/Abis	Ethernet/IP	IP/ MPLS/ Ethernet

.

The different generations of mobile technologies were developed one after the other in order to acquire better scalability, higher speeds, lower cost and increased network capacity which combined to give the user a better QoE and operators a lower cost. IP based backhaul is also a result of all these requirements. One of the requirements of IP-Ethernet-based MBH networks is differentiation of traffic based on class and even prioritizing within the class. This is particularly important for delay sensitive applications that need higher priority such as Voice over IP (VoIP). Packet based networks have flexible bandwidth planning and management unlike the legacy networks' fixed bandwidth; which provides a chance for more differentiation on the traffic flows and assists in getting income from policy based Quality of Service (QoS) provisioning. [13]

In Figure 5, the Metro network uses carrier Ethernet to carry IP, MPLS and Virtual private Network (VPN) services from the mobile core to the backhaul network. LTE uses a flat IP architecture that is a network where nodes reach other nodes using IP connectivity in a much simplified data architecture. In IP based networks, different types of traffic can be transported over a common IP/MPLS network. [17] This is particularly advantageous in that the core network only needs to check the MPLS label from the protocol stack.

Following the general trend towards all-IP networks, the mobile core network (LTE Evolved Packet Core (EPC)) is changing as well to a more flatter architecture with IP transport as opposed to the tiered 3G architecture. In addition, the functions of EPC can be relocated in a distributed manner to the Metro transport network. This reduces latency by pushing some functionality towards the edge and simplifies the core. The distributed architecture places more responsibility on the backhaul to provide higher level IP routing functionality and security towards the edge. [13]

3.3 Multiprotocol Label Switching (MPLS)

MPLS is a technology that does switching using labels rather than IP addresses. It can help in sending packets that are destined to the same node to travel in a different route by assigning them to different labels. This plays a crucial part in load balancing and TE. MPLS facilitates TE by not using only the shortest path -contrary to pure IP networks- but using labels to select arbitrary paths to avoid inefficient use of links, decreasing load on the shortest path.

MPLS brings the idea of label switching from technologies like Frame Relay and ATM to IP network. The advantage of using label switching lies in the formation of path along which the packet goes. This stable path makes it easier to identify different types of traffic which in turn helps in identifying traffic types that need to be serviced fast. The MPLS header is placed between the IP header and frame header. MPLS used tunnelling to transfer labeled packets from ingress Label Switched Routers (LSRs) to egress LSRs. The tunnel is called Label-Switched Path (LSP). The router from where the LSP starts is called ingress router and the end of the LSP where the label is stripped off of the packet and processed is called egress router. Between the ingress and egress routers there are intermediate routers. [18]

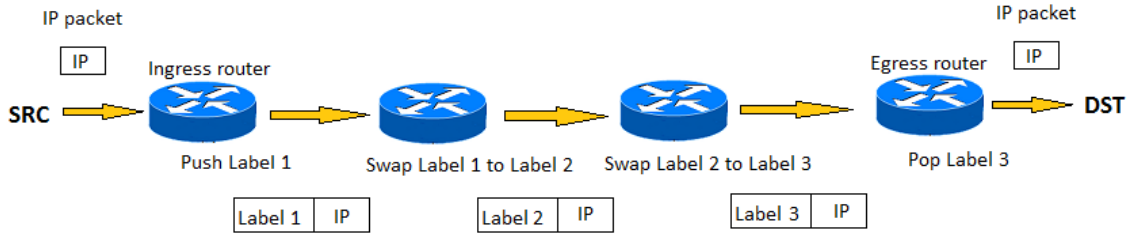

Figure 6: MPLS label swapping

At the ingress LSRs IP packets are classified into classes then an MPLS label is added to each packet according to their classes and the LSR forwards them to next hop using the labels as index. [3] The intermediate routers take incoming labels and change them to out labels (swap labels) that are advertised to them as the outgoing labels to a particular destination as shown in Figure 6. The MPLS header is 32 bits long and the label is 20 bits long. There are 3 experimental bits. The S bit tells whether this label is at the bottom of the label stack. The 8 bit Time to Live (TTL) serves to avoid routing loops. The path of the packets is predefined between ingress and egress by MPLS label. [19]

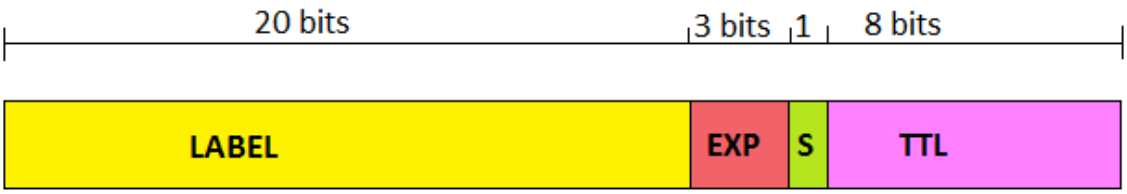

Figure 7: MPLS label

Resource Reservation Protocol for Traffic Engineering (RSVP-TE) is a signaling protocol for MPLS used by routers to set up LSPs taking constraints into consideration and for applying TE capability as well as QoS. It is commonly used by operators to aggregate the same QoS level traffic into the same tunnel. The Label Distribution Protocol (LDP) is another signaling protocol which is not used much today. It is a simpler protocol but cannot apply QoS to LSP. LDPs are mechanisms by which one LSR informs another LSR of the label to be associated with a particular Forwarding Equivalence Class (FEC). FEC consists of group of packets with similar or identical characteristics which are forwarded with the same treatment over the same path. Multiprotocol Border Gateway Protocol (MPBGP) can also be used for setting up path based on the routing information even if it is not a signaling protocol. Lastly, LSPs can also be configured statically. MPLS tables have incoming labels, outgoing labels and the next-hop router on the LSP. The ingress and egress routers also have similar arrangement except that they will have one label either push or pop. [18] In MBH there can be multiple LSPs assigned in order to carry packets. MPLS is often chosen for its ability to carry a diverse range of traffic over a common MPLS transport and due to the support of QoS.

In MPLS-TE a tunnel has properties it requires in the links it is going to traverse. These are attributes for which the tunnel has an affinity and they are configured for the tunnel. They are called the tunnel's affinity bits. Affinity is used to control the path selection process of the MPLS-TE Tunnels. The tunnel's affinity bits (tunnel id) and affinity mask must match up with the attributes assigned to the link so that the tunnel chooses these links over those which do not match. The attributes are assigned using a 32 bit attribute flag that can indicate up to 32 different properties are possible on the link . The affinity mask is the mask that helps to identify which link attributes (which bits) need to match. [20] With include and exclude affinities -which are affinity constraints-, it is possible to include or exclude links or tunnels for forwarding decisions.

3.4 MPLS Layer 3 Virtual Private Networks (MPLS L3 VPNs)

A Virtual private Network (VPN) is a network that uses public infrastructure such as Internet or shared network in order to provide remote users with access to a private Intranet. It often provides similar security and management benefits as being in the private network based on policies. MPLS enables the provisioning of provider-

assisted VPNs to connect remote sites together at the IP layer (also called Layer 3 VPNs). Each VPN contains one or more Customer Edge (CE) devices and the CEs are attached to one or more Provider Edge (PE) routers [21]. Using a LSP, the VPN creates a tunnel between the parties where information need to be exchanged i.e a virtual point-to-point connection is established by tunnelling that emulates a leased line. Users can connect securely to the private network or part of the network through A VPN tunnel. CE routes send datagrams towards the PE routers and the PE routers examine their IP headers and route them. With site-to-site VPNs, parts of an organization in different areas can communicate through a virtual common network. The VPN is preferred over frame relay and ATM networks due to its decreased cost. Even when there is overlapping address space between two VPNs they don't communicate and remain distinct.

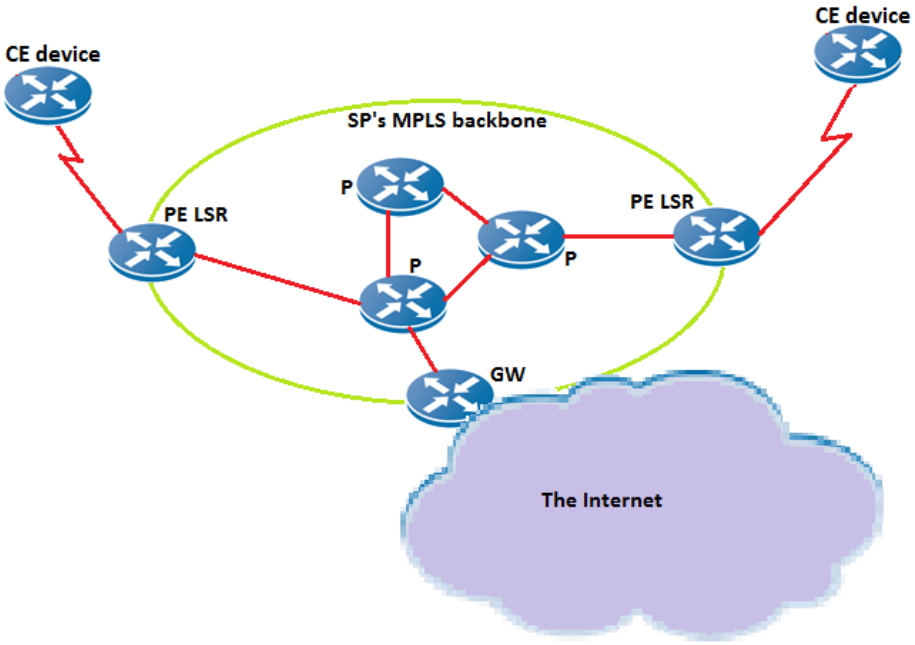

Figure 8: Simple MPLS L3 VPN connecting site A and B

VPNs can be created using MPLS backbone. In this case, each route attached to a VPN is assigned an MPLS label. An MPLS VPN is a variant of layer 3 VPN. The service provider has PEs that are LSR; the PE routers have a default forwarding table and multiple VPN Routing and Forwarding (VRF) tables. Each VPN possessing its own VRF allows the separation of private networks and address space overlapping mentioned earlier. When a packet arrives at PE router, the routing and forwarding table is checked for its destination and for information on how to route the packet. If it is not a VPN related address, it will be forwarded based on the default forwarding table. For site-to-site VPNs the connection from the sites

(CE) to the PE LSRs is made and the PE takes care of the routing between the sites that belong to the same VPN. Sites within the same private network have the same forwarding table and can communicate to each other via the PE that is attached to that particular VPN. In Figure 8, the P-routers do not keep any state information. They simply perform MPLS routing. The routing information is only stored by the PEs that serve the VPN in question, thus freeing the other SP routers from the management of other VPNs they do not directly serve. Border Gateway Protocol (BGP) is used by the SP in order to exchange routing information (VPN routes) between the PE routers and distribute labels to all PE routers needing them. When there is overlapping addresses in different VPNs, BGP must receive additional information in order to be able to distinguish routes for the same address and not omit one or more of them thinking they are routes to the same destination. The information used to perform this separation is provided by an 8 byte number called a route distinguisher (RD). It is prepended to the Internet Protocol version 4 (IPv4) address to form a BGP VPN-IPv4 address. [21]

3.5 Pseudowires

Pseudowire is a mechanism for emulating telecommunication services such as E1/T1, ATM, Frame Relay or Ethernet over Packet Switched Network (PSN) using the likes of IP or MPLS. Pseudowires carry layer 2 traffic. Pseudowire Emulation Edge to Edge (PWE3) encapsulate service related information arriving at an ingress port and transfer them across an IP or MPLS tunnel. It also manages other aspects related to the service such as signaling at the boundaries of the pseudowire. For the customer edge, it appears as if it is using a native service and does not know about the emulation. PWE3 is the pseudowire emulation edge-to-edge working group which defines how Layer 2 traffic is carried across the network. [11] The emulation is done for only necessary functionalities for the service intended. When packets arrive at an ingress port, service related information is encapsulated by PWE3 and sent along the PSN path. PWE3 may also manage the timing of this information and how well the services are emulated. [22]

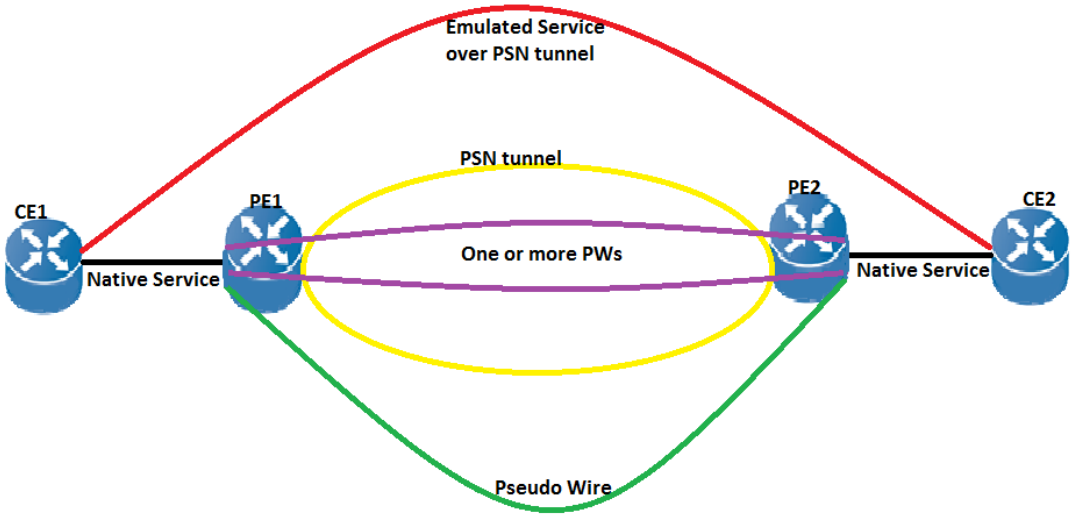

Figure 9: PWE3 network Architecture for point-to-point pseudowires. [22]

As depicted in Figure 9, native PDUs are sent from CE to PE that pass through an encapsulation layer at the PE and are sent over the PSN after being encapsulated in pseudowire-PDU. The PEs provide pseudowires for their respective CEs in order the CEs to communicate over the PSN tunnel which forms a path for the pseudowires. The pseudowire provides an emulated physical or virtual connection between the CEs. The PE on the other end removes the encapsulation to change the payload to a native format to transfer it to the its CE. [22]. In PSNs, packets are usually fragmented and reassembled at the destination, and therefore subject to delay and packet loss. Different services have varying levels of adaptation to these characteristics of the PSN.

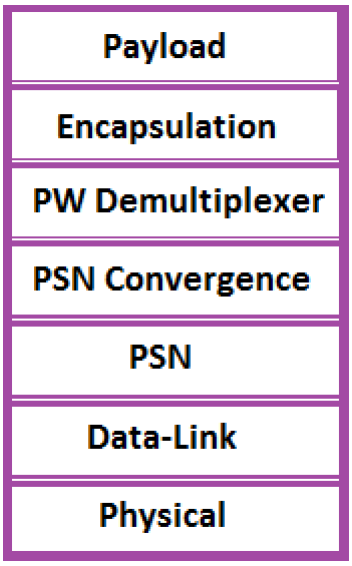

Figure 10: PWE3 protocol layering. [22]

The protocol layering model to support pseudowire is as shown in Figure 11. The encapsulation layer that is associated with payload convergence, timing and sequencing is provided by the PWE3 while the other layers are provided by the tunnelling. The model is planned so that the pseudowire definition is independent of the underlying PSN tunnel. The payload or various payload types are transported over the Encapsulation layer which could be empty if all necessary information for the reconstruction of the payload is present in the payload itself. The encapsulation helps adapt the payload to the pseudowire demultiplexer that takes the payload over the PSN. The pseudowire demultiplexer is the value that assists in finding multiple pseudowires over a single PSN tunnel. The PSN convergence layer improves the PSN to meet its service requirements so that the pseudowire is independent of the PSN type. It is empty when the PSN conforms to pseudowire's service requirements for PSNs. The payload goes to the CE through a physical interface. A pseudowire must be setup before an emulated service is started and stopped when an emulated service is no longer needed. [22]

3.6 Challenges of MBH

MBH demands a strict QoS and OAM capabilities. In order to grab the opportunity MBB brought about, operators need to provide more capacity and coverage. The dynamic traffic flows and QoS requirements need to be met and this is possible with highly evolved technologies such as Evolved High-Speed Packet Access (HSPA+) and LTE . Therefore, the backhaul needs to evolve from TDM to packet based architecture as well. The solution needs to be cost effective. This however is not without a challenge because the legacy network has been built expensively and the services are still widely used. The migration needs to happen without affecting existing services. Some solutions such as using pseudowires to carry TDM traffic en-

capsulated over packet based transport are used to tackle part of the problem. [13]. As the evolution of RAN towards an all-IP network takes place, service providers need to migrate towards LTE and they desire to have a high quality backhaul which transports their traffic. The solution adopted by the MBH should facilitate easy migration towards the next generation of technology. The challenges of the MBH currently arise due to the complexity and overhead created by the need to support heterogeneous technologies simultaneously.

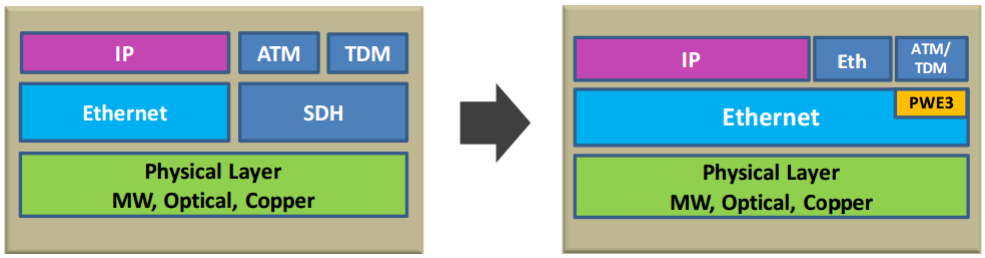

Figure 11: The migration from hybrid to pure packet protocol stack. [13]

High bandwidth consumption results from the increasing demand for delay sensitive applications such as video conferencing and bandwidth hungry video and image applications. According to Cisco's global mobile data traffic forecast update, mobile data has increased 81% in 2013 reaching 18 exabytes of traffic which is 18 times the size of the entire global Internet 13 years earlier. Smart phones and smart devices account for the majority of the mobile data traffic growth. 4G connections account for 30% of the mobile data traffic even though they represents only 2.9% of mobile connections. Mobile data usage is expected to grow even larger within the next few years. [23] Backhaul is posed with a challenge because capacity growth must match the increase in consumer demand while containing the costs of meeting that demand. The OPEX of MBH increases due to the exponential growth of mobile data traffic. However it is not the only cost as the space, power, cooling and hardware costs are increasing as new technologies are added, since there is still a need keep legacy services [11].

The backhaul technology is transitioning from legacy technologies such as T-1 service to Ethernet service. [24] The MBH network has a significant influence in the end-to-end quality of mobile network as it comprises a large portion of the network between the base station and the core. MBH's share of total costs is a significant amount that can be between 10% and 40% of the total network related mobile operator cost. Legacy backhaul networks are serving mainly 2G or some 3G mobile networks however the radio technologies are progressing fast which calls for the need to evolve the MBH. [25] The 2G-serving backhauls are usually based on TDM transport technologies such as Plesiochronous Digital Hierarchy (PDH) and Synchronous Digital Hierarchy (SDH)/Synchronous Optical Networking (SONET) which are of fixed bandwidth suited for TDM-based circuit services. In the 2G era, there was a Multiservice Transport Platform (MSTP) that was used to enable IP services to

be carried over transport networks. However, it was not an all-IP solution because even if its user interface may be IP-based its core is TDM-based with a fixed tunnel to carry packet services. It was not cost and transmission efficient and suffered from poor scalability processing variable length packets, bursty IP and Ethernet services standing in the way of mobile services development. This led to the development of Packet Transport Networks (PTN). [26]

The 3G serving backhauls include ATM equipment which facilitate the efficient use of bandwidth than mere TDM. New backhaul networks contain packet technologies. Packet technologies are incorporated in the legacy networks due to the proliferation of MBB. PTNs combine the advantages of packet technology and SDH. The PTN uses packet switching which provides a good packet transfer service. The PTN has strong OAM delivered from SDH and carrier-class security protection services. For example, MPLS-Transport Profile (MPLS-TP) has high level of OAM capability that can be compared to the strong OAM offered by ATM and SDH and it incorporates enhanced protection and synchronization. MPLS-TP therefore is suited for carrying IP-based MBH services. The future is for packet based MBH because packet based transport solutions of similar capacity are much cheaper than legacy solutions and consume less power. [25]

In Global System for Mobile Communications (GSM), voice was the main traffic type with 16 Kbit/sec per voice channel, hence transport capacity required per base station was minimal (2 to 8 Mbit/sec.) [25]. Therefore, in GSM networks, it was easier to build a backhaul that can carry the traffic the base stations are providing. TDM transports however are not efficient in carrying very heavy data traffic because E1, ATM and SDH protocols do not provide a scalable solution capable of supporting the volumes of traffic offered by variable data rates. Ethernet and MPLS are better suited for this service [13]. Mobile network capacities are evolving with new network generations and backhaul capacity needs to grow as well so that it is not the bottleneck for QoS. In order to bring about this MBH progress whilst keeping OPEX in check, network simplification and automation is necessary. [25] SDN as mentioned in chapter 2, aims to bring the simplification of the network architecture and automation of management and operation.

The migration should happen gradually. MPLS based transport technologies such as MPLS-TP and IP/MPLS are preferred in the backhaul because they support strong pseudowire functionality that can carry legacy technologies encapsulated in the packet technologies, thus assisting gradual migration. As TDM is removed in the process of converting to all-IP, synchronization and timing problems arise which is addressed by synchronous Ethernet technologies developed for this purpose. [13] Services are moving away from TDM voice towards IP-based voice and data. Interfaces are changing from E1 to Fast Ethernet (FE) and (Gigabit Ethernet)GE for increased bandwidth. Demand on the 3G network is mounting for better service and high QoS. A MBH network needs to fulfil the requirements needed by an all-IP network. It must have efficient statistical multiplexing and differentiated QoS. It must

not drop traditional transmission characteristics such as end-to-end service management, hierarchical operation, OAM and carrier-class protection. PTN technology with connection-oriented packet technologies supports all these solutions. It carries both carrier-class Ethernet services and TDM ATM services making it a solution for IP-based MBH networks. Example of technologies for PTNs are MPLS-TP and Packet Backbone Bridge Traffic Engineering (PBB-TE). MPLS-TP is the de-facto mainstream technology for PTNs. PWE3 help PTN in carrying multiple services. PTN provides access link, network-level and device-level protections. The OAM mechanism of PTN is used to achieve timely fault detection and this is a requirement for carrier-class protection. [26]

Due to the fast changing network services and the quickly growing demands placed on the network, network adaptation must be done quickly. SDN provides the tools to bring about the necessary network agility independently or closely working with Network Function Virtualization (NFV). NFV is running applications and network functions on virtual machine-based standard computing servers shared with other network functions. It runs network functions in software rather than on proprietary hardware appliances, consolidating network elements such as servers and storage leading to a fully virtualized environment [27]. NFV scales the network functions flexibly with demand and places instances of network functions in the network where they can be most efficiently used based on current and forecasted demand. This optimizes the network, the functional flexibility however, depends on SDN for dynamic adaptations of the interconnections between these dynamic functions, within the components of the functions and between the functions and end users. That is, SDN assists in the flexibility of NFV. Data centers running IT applications relied on VLAN and IP subnetting before, leading to scalability and flexibility limitations. The network was being the bottleneck for real agility in the data centers, because it was not easy to adapt to these changes on the network sufficiently fast. SDN addresses this problem and tries to bring the same level of flexibility to the network as is available for compute and storage resources in the data centers. [28]

3.7 Chapter Summary

MBH is the part of the operator's network that carries traffic from the cell sites to the network core. It constitutes a very large portion therefore, it must evolve with the RAN to avoid being the bottleneck. IP based backhaul solutions are preferred in the 4G networks as they provide differentiation based on QoS and Class of Service (CoS) and flexible bandwidth planning. MPLS based transport protocols are especially beneficiary for their support of pseudowire functionality which emulates different generation telecommunication services over packet switched networks assisting in the transition to an all-IP network. The current MBH is heterogeneous having different technologies to co-exist thus hugely complex. SDN-based automated service provisioning can offer resource management in the MBH. An all-IP network in MBH makes the introduction of SDN easier because of it simplified architecture. Then

when SDN is introduced, it assures the centralized control of the modules in the MBH. Operators strongly desire to keep the CAPEX and OPEX in check in MBH while maintaining the QoE they promised on the SLAs. SDN makes the forwarding plane cost efficient helping with minimizing of CAPEX and the automation and adaptation characters of it minimizes OPEX.

4 Traffic Engineering Optimization in MBH

As discussed in chapter 3, MBH constitutes a large portion of an operator's network. It needs to be properly managed and traffic paths need to be engineered meticulously. This is important to prevent MBH from being the bottleneck for network performance serving the continuously increasing demand for heavy data traffic. Resources need to be provisioned efficiently while accounting for constraints on the network. Network resources are expensive thus it is not feasible to provide dedicated connections to each and every device; rather, network owners attempt to manage with the available resources for as long as possible and add as few new ones as possible. This chapter presents methods and techniques used by operators, as well as the challenges they face in managing congestion and extracting the maximum value possible from their capital investments. Additionally, different TE mechanisms are discussed and their advantages and shortcomings are highlighted to show how TE has been implemented in operator networks. The use of SDN in TE for MBH is also outlined in this chapter.

4.1 Overview of TE

Traffic engineering (TE) is a set of processes or mechanisms applied in order to facilitate the efficient use of bandwidth in a network. TE can be used to define paths to prevent some links being overloaded while the others are left underutilized. QoS is a traffic delivery methodology that must satisfy special requirements. TE on the contrary is concerned with traffic management of the whole domain rather than for some "chosen" applications. However it is able to differentiate based on classes and quality and provide separate links for different types of traffic accordingly. The distribution details about the links more than the metrics found from the routing protocols is required for TE. These information include such information as the available bandwidth and over-subscription rates. [19][29]

End-users expect the performance they have been promised by the service provider for the amount they pay. Continuous performance evaluation and optimization is needed with the resources and the traffic in order to meet the performance requirements economically and reliably. TE is network engineering that tries to solve the challenges posed by the performance requirements of a network. It tries to resolve these issues by applying scientific principles and technology to assess the traffic and provide mechanisms and policies to bring about the solution. The traffic is assessed by using measures such as delay, delay variability, packet loss and throughput. TE mechanisms need to be well defined for known requirements and readily adaptable to future demands. [3]

In early telephone networks, TE was not very efficient as it used static hierarchical routing which did not take the state of the network or time of day into consideration. [3] At that time, each call is allocated a dedicated bandwidth for the

duration of the call, and the call is blocked if there is not enough bandwidth. The fixed routes are always there, optimized to accommodate busy-hour traffic, even if it is not busy hour. This is a very expensive provision option and lacks flexibility. Dynamic routing by which the resource is allocated at run time and periodically updated helped the lack of flexibility. Dynamic routing can be based on time, state and event [3]. The introduction of dynamic routing has improved TE by considering these factors that are constantly changing in the network.

In the Advanced Research Projects Agency Network (ARPANET) (the first packet switching network), TE optimization has always been there and efforts to improve it existed. It started with recognizing the importance of dynamic state based routing. This adaptive routing by which the route from source to destination depended on the state of the network, helped improve performance by contributing in congestion control. However, it was not optimal as it caused oscillations by responding to the change in the state of the network too quickly. The Internet started as a best effort service thus traffic management and QoS based on class were not major parts of the development. It adopted adaptive dynamic routing algorithms from the ARPANET to determine the path packets take en route to their destination. The routing protocols used algorithms that calculate the shortest path based on static and dynamic link metrics. The early Internet routing protocols did not take factors affecting TE optimization into consideration while making routing decisions. This causes suboptimal utilization and worse congestion. The inadequacy of the legacy Internet for TE, motivated path oriented technologies such as MPLS. Virtual-circuit (VC) connectivity that is used in technologies like frame relay and ATM, has played a role in TE by performing path optimization. This is done by rearranging the VC so that a VC on a congested link can be switched to an optimal link. [3]

Constraint-based routing is routing while considering constraints, that arise either from the network such as resource availability or from different management and service oriented policies. Its aim is identifying a path, that is in accordance with a set of constraints such as delay and packet loss rate. This enables a routing paradigm that is responsive to the traffic demand, playing a great role in TE as it is more considerate of the network state than traditional routing. The methodology is strengthened by path oriented technologies such as MPLS. In QoS, the requirements could be delivery of packets without loss and with minimum delay; other requirements would be considered depending on the available technology and user needs. Constraint based routing is different from QoS routing as it is applicable to traffic aggregates and flows rather than individual traffic flows. The information TE gets about the links will be combined with the standard metrics from routing protocols to find the "best" path, a process called Constraint Shortest Path First (CSPF). CSPF is similar to SPF which is used in such protocols as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS), however it considers the constraints added by TE. That means the links which fall short of satisfying the constraints will be pruned before the calculation of the shortest path.

The downside of CSPF is the CPU power and time it takes to calculate the shortest path; which poses a shortcoming for TE using this algorithm. [3] [19]

4.2 TE Methods

TE optimization can be divided into two steps on a high level; performance evaluation and performance optimization. Performance evaluation is done with the aim of checking whether the network performance requirements are met. It can be done by analytical, simulation, empirical methods or the combination of those; but in practical networks it is rather complicated and different techniques are needed before doing the analysis. It reveals if there is a network failure, suboptimal use of network and helps in finding ways for TE optimization. [3]

TE optimization is done by monitoring the network in order to create optimal network, prevent and relieve network congestion related issues. TE optimization control can be done pro-actively and/or reactively. TE optimization may involve increasing link capacity or adding links, deploying additional network elements, adjusting parameters, routing metrics and adjusting traffic management. If there is a need to improve the whole network architecture it then involves non-real time network planning process. Network optimization is not always done to correct problems; it can also be perfective action taken to continually improve network performance. [3]

In order to come up with traffic optimization means, we first need to know what the performance requirement of the network is; we also need to know the set of policies and constraints related to them. Tools and mechanisms for the measurement of traffic and the monitoring of resources and traffic are necessary. We then need to formulate what the TE problem is. A configuration management system can complement all these by helping in administration of how the tools and policies are implemented. Routing analysis to know what are the paths the routing protocol selects, and how it assigns the traffic is a crucial element of TE optimization [3]. This is because in TE we need to know the effectiveness of resource usage. Having said that, in SDN architecture, the network topology model can be easily found from the controller making the routing analysis easier.

RFC 3272 states four phases of a generic process model for Internet TE that must be performed by a TE system continually to optimize the performance of an operational network. The flow chart shown in Figure 12 illustrates these phases. The first phase is defining control policies to oversee the operation of the network. The policies are shaped by factors such as business model and constraints on the network. The second phase is getting measured, estimated or extrapolated data from the operational network. The measurement needs to be done systematically knowing the answers to the questions why measure, what to measure, how to measure, where to measure, how often, what measurement accuracy, cost of measurement and the likes.

The third phase is to analyze the network state pro-actively or reactively in order to characterize the traffic load and identify if there is any problem or inefficiency in the network or in the traffic distribution. Modelling and simulation assist in the second and third phases. In analysis, network simulation is widely used because certain aspects of analysis can only be efficiently done with simulation. The fourth phase is selecting a set of actions from a solution pool to optimize the performance of the network. Simulation plays a vital role in the phase of optimization especially in network planning to explore how to make the network evolve and grow efficiently. [3]

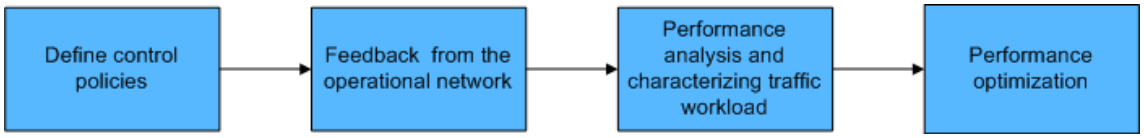

Figure 12: Traffic engineering process model.

TE in the Internet uses different traffic markers and classifiers and policing mechanisms. Queueing methods are also included in TE such as class based queueing, First In, First Out (FIFO) queueing, priority queueing, Fair Queueing (FQ) and Weighted Fair Queueing (WFQ). Furthermore, there are many Internet Engineering Task Force (IETF) projects that are relevant to TE because they intend to evolve the IP architecture to support preferential or differential treatment of certain types of traffic. One example of these is MPLS Traffic Engineering (MPLS TE). MPLS has TE in its nature; it is a technology that integrated TE capability in Layer 3 and enhancing the efficient use of bandwidth. MPLS TE is advantageous as it combines the TE capabilities from ATM with CoS differentiation of IP in order to treat different types of traffic in different manner [29]. Other examples of these projects include Integrated Services (Intserv) and Differentiated Services (Diffserv).

4.2.1 Integrated Services

The Integrated Services project has developed IntServ model for guaranteeing per-flow end-to-end QoS. This architecture supports per-flow traffic classification and scheduling algorithm at devices. This model contributes to TE by requiring resources to be reserved before a traffic flow starts to guarantee the required QoS. Intserv has the following additional components that are depicted in the figure below to augment the best-effort model. In IntServ, resources are reserved by each flow and flows having highest priority are served with the resources. [3][31]

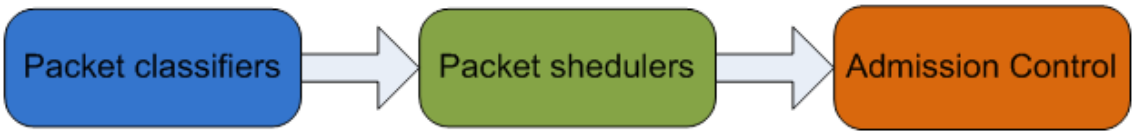

Figure 13: Components of Intserv

Packet classifiers are components that find flows intended for a special service, packet schedulers schedule services to different packet flows and admission control is the component that controls if there is available resources for the flow which is about to come. Intserv services could be guaranteed services, for applications that cannot tolerate delay and must know end-to-end packet delay bound. They could be also controlled-load services for applications that tolerate some delay but cannot endure traffic overload. These applications are provided with a lightly loaded network even if the actual network is in overload condition. Even though it guarantees good QoS for each flow using resource reservation, the problem with IntServ is that as the network scales up, it makes the controlling of the reservation rather complicated as every router must support IntServ and keep many states. IntServ defines the semantics of the QoS with flow specs that tell what the reservation is for, that is, they define the traffic specification and the QoS requirements the traffic has. Resource Reservation Protocol (RSVP) is the mechanism for signaling of QoS requirements from end systems to routers. It was originally developed as a signaling protocol within the Intserv. Originally Intserv was planned to be used for every flow in the Internet, thus not very scalable. [3]

4.2.2 Resource reservation protocol (RSVP)

RSVP is a resource reservation protocol from end systems to the network requesting the latter to reserve relevant resources to satisfy the requested QoS requirements of applications. RSVP however does not define or provide the QoS itself. It simply transports the IntServ objects and transfers them to the traffic control component at each node along the path. It is not a routing protocol but it works with routing protocols for routing RSVP messages. [32] The source node sends a PATH message to the receiver specifying the characteristics and format of the traffic and this message is transferred through intermediate routers along the path according to the routing protocol used. The receiver responds with RESV message back in the opposite direction along the path of the PATH message when it receives the PATH message. RESV message includes flow descriptors used to request resource reservations. If any intermediate router rejects the request, it sends an error message to the receiver and the signaling process terminates. If the request is accepted, the resource is allocated for the flow or aggregation of flows. [3]

4.2.3 Diffserv

Another IETF project related to TE is Diffserv that was proposed to lessen the scalability issue of the Intserv model using the concept of traffic aggregation instead of a per flow basis. It works with aggregated traffic categorized by behaviour and packets are scheduled and forwarded at each device on a per-class basis. The services with critical requirement are served with low latency and others with for example best effort. A 6 bit Differentiated Services (DS) field has been defined in the IP header in the Type of Service (ToS) field of. This field tells what kind of treatment the packet should receive at a node. The number of classes is limited in

the DS. The amount of state information is proportional to the number of classes, unlike the number of application flows as in the case of Intserv. This gives Diffserv a scalability advantage over Intserv. [3]

Diffserv deals with TE issues on a per node basis and other TE capabilities are needed for a better service quality in the whole network. The maximum queuing delay in IntServ can be foreseen based on the amount of resources reserved. In DiffServ however, a common network resource is used by all flows belonging to the same QoS class. Therefore the end-to-end queueing delay is also based on the amount of traffic from the other flows in the same class. It can be taken that IntServ has better QoS guarantees than DiffServ but this advantage over DiffServ comes to effect only when the total amount of traffic and the resources are equal. Moreover, in DiffServ networks, as flows share the same network resources it benefits from the statistical multiplexing gain. Therefore, DiffServ networks exhibit smaller queueing delay for more flows than IntServ. [31]

DiffServ would guarantee QoS to more traffic if admission control function is applied to both DiffServ and IntServ. The advantages of IntServ and DiffServ could be combined to have a system with good QoS guarantee and scalability. Integrated Services over Specific Link Layers (ISSLL) working group proposed that IntServ be used at the network edge where there are less flows and DiffServ be used at the core due to its scalability. The boundary routers between those domains choose the mapping between IntServ flows and DiffServ classes. The DiffServ class must be selected taking into consideration the type of IntServ service requested for the application. [3] [31] [33]

4.2.4 Resource Reservation Models

After path computation, the bandwidth needs to be allocated to different Class Types (CTs) and the bandwidth the CT or group of CTs can use is called bandwidth constraint (BC). BC is important in determining the bandwidth that is available for each class on a link at each priority level. The Maximum Allocation Model (MAM) and the Russian Dolls Model (RDM) are DiffServ bandwidth constraint models. The MAM maps one CT with one BC, meaning the link bandwidth is divided among different CTs. MAM enforces strict separation between the bandwidth allocated to different CTs. The bandwidth allocated for one CT can not be used for another CT. In MAM model, unused bandwidth can not be shared between CTs and it is wasted rather than carrying other CTs. Its advantage over RDM is simplicity in understanding and managing. RDM bandwidth allocation model allows sharing of a bandwidth across different CTs improving bandwidth efficiency. [34]

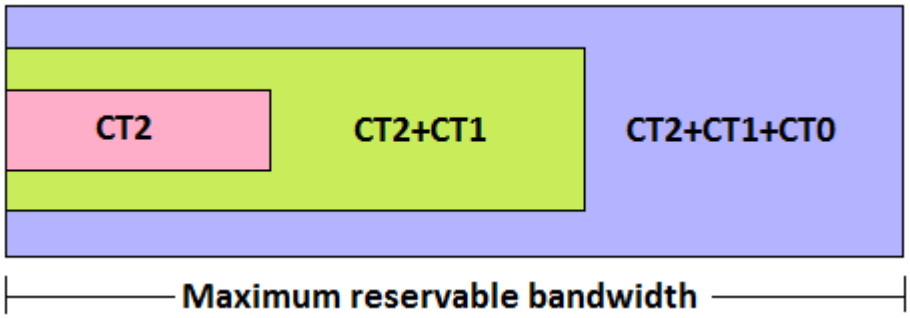

Figure 14: the Russian dolls model example with three class types.(RDM).

In Figure 14 CT2 is the highest QoS requirement traffic and CT0 is the best-effort traffic. CT2 will get a fixed amount of bandwidth that is not shared and also shares traffic with the other CTs hierarchically. Bandwidth wastage is decreased because as long as it is not being utilized it is available for any traffic class. For example, when we allocate for real-time CoS, the unused bandwidth will be used by other classes making it cheaper provisioning. The disadvantage relative to MAM is because it shares bandwidth resources, there is a need to assign different priority to the different paths from various classes to ensure each CT -even when it is not high priority- gets its share of bandwidth. That means RDM requires extra planning and configuration. [34]

4.3 Congestion Management

Congestion occurs when the arrival rate of packets exceed the output capacity of a node. Thus congestion happens when there is a shortage of bandwidth. The shortage of bandwidth can be caused by inefficient traffic management causing inefficient traffic distribution. In IP networks, usually the shortest path is chosen even if the other routes are underutilized. Congestion may force packets to be delayed -or even worse, dropped- causing a degradation of service quality for end users. Minimizing congestion is a vital part of TE. QoS and CoS are used due to finite resources and a selection is needed to be done in order to prioritize high priority packets. Capacity management constitutes a crucial aspect of TE and helps to ensure the network meets current and future performance requirements. During this process it is possible to find what the constraints of that particular network are, and by studying traffic, we can see if the network is now or in the future prone to congestion. We do this by forecasting traffic load and providing nodes and links with a test load or taking any other necessary actions [3]. Performance requirements may change over time making capacity management a constant challenge requiring a continuous congestion management process .

Congestion management policies can be reactive or preventive. Reactive policies are recovery mechanisms when known or discovered congestion problems occur. Preventive policies are proactive congestion avoidance mechanisms using forecasted traffic demand and distribution of probable congestion problems in the network. [3]

TE in the Internet includes congestion control mechanisms such as Random Early Detection (RED), Explicit Congestion Notification (ECN) and TCP rate control. Congestion notification to the end systems is imperative so that they can make any modification needed, knowing that the transmission they planned did not happen. Implicit congestion notifications are based on an acknowledgement (ACK) expected after each packet is transmitted. If ACK is not received the source host assumes congestion and may decide to retransmit and if the receiver notices a retransmission it assumes congestion happened and its ACK message was dropped. The implicit notification is not suitable for protocols that do not use ACK. Therefore, ECN is needed. This is a technique whereby a mechanism on the device on which congestion happened, informs both source and destination about it. ECN is implemented mainly in frame relay and ATM switches. Congestion avoidance mechanisms slow down the transmission speeds of some flows in order to avoid congestion. RED's use impacts the windowing mechanism of TCP. It drops some packets randomly and when the TCP source does not receive ACK it halves its TCP window reducing the throughput of the flow avoiding congestion. RED treats all flows as equal but it is possible to introduce different drop profiles to each class of traffic. However having different drop profiles for each class might prove complex. Routers put packets that arrive faster than its processing speed or outgoing link rate in queue in order to process and transmit them. [19]

Routers have finite amount of buffer memory to hold queues therefore, they will discard the packets when this memory is full. Preferential treatment to packets is done so that devices service different queues at different rates. There are different variants of queueing strategies. FQ is a mechanism that ensures every queue with packets to be transmitted, has an equal share of the bandwidth per bit basis and makes sure no flow uses more than its fair share of bandwidth. FQ has some shortcomings such as not supporting preferential treatment thus not supporting QoS. Additionally, when packets have different sizes it becomes less effective because it assumes all packets to be of same size. Due to that it tends to prefer flows of larger packets. Another variant of queueing is WFQ. WFQ did not depend on FQ for development but it solves some shortcomings of FQ by allowing different scheduling priorities. Flows will be put in queues according to their priority. The queues are transmitted based on the weight they are given. If the packet units are bigger than the weight it will not be transmitted during that round. [19]

WFQ provides different service shares to different queues creating a technique to favour high priority queues. It provides class or flow based queueing. It also supports fairness in case of variable packet lengths; its limitation being complexity. The complexity makes it challenging to apply it to high bandwidth flows. Round-Robin (RR) is another variant of queue servicing and it services each of the queues without priority in a circular fashion avoiding bandwidth starvation of any queue. Weighted-Round-Robin (WRR) adds weights to RR to modify the rate at which each queue is serviced providing differentiated QoS. WRR like RR is difficult to implement in its original form because packets need to be fragmented to blocks that

are proportional in size given in bits to the weight applied to the queue. This is difficult because reassembling of packets will be impossible later. [19]

Weighted-Deficit-Round-Robin (WDRR) method is similar to WFQ, but rather than not transmitting when the packet units are greater than the weight at that round, it will transmit and it will increase the deficit of weight against that queue for the next round by units of packets exceeded in the previous round. It is easier to implement than WFQ, which makes it applicable to very high speed interfaces. It does have shortcomings such as not providing protection against a bad flow and not offering accurate control against delay and delay variation of packets in the queues. Priority Queueing is another type that gives better control over latency and jitter. Priority Queue will include prioritized classes of traffic which are delay sensitive. This also has its flaws because it might make other queues not receive any share of bandwidth at times, if not controlled by an upper bound bandwidth. The different types of queues are used in combination in order to overcome their limitations to provide good customer service. [19]

Policy writing is very important in traffic management. How policy writing is approached and implemented affects the scalability of the network. Routing policy is set to tell how the devices in a network behave during reception and forwarding of traffic. For example, which path to take and which sub-network to choose. Policy routing is routing using policies as constraints or filters instead of merely using default routing tables. With policy routing, a path can be preferred for some traffic even if it is not the optimal path as long as it supports business needs. Policy plays a great role in defining how to meet the business needs of an organization. It is also imperative for an organized way of transferring information. Well written policies help network engineers create configurations for new devices or apply policy while implementing a new protocol. Policies can be applied to packets or to a particular protocol. Policies include actions such as permit, deny, discard that can be applied to routing. [19]

4.4 Challenges in TE

TE optimization needs to be done continually with the goal of improving network performance. Controlling the routing and managing the resources is a very complicated task and the management overhead is high. Often there are too many network elements and they make local decisions thus network operators cannot control each and every device easily. In order to tackle this problem there is a need to develop powerful management systems for the vendors. These management technologies are very expensive to develop and operate. For example, The INM in Section 5.1.1 -which can be used to automate TE- has its own separate department with many experts which is evidence of how expensive NMSs are. They also need to be continually developed to ensure their meeting of emerging performance requirements. Performance optimization in a backhaul is a very important challenge to tackle be-

cause performance requirements from the operators are becoming multifaceted and at times even contradictory based on the demand of end-users. Thus, a backhaul service and product provider needs a powerful network management system.

The challenges in TE are related to formulating the problem, measuring and estimating relevant network state parameters, characterizing the state of the network and finding effective solutions to optimizing network performance [3]. Formulating the problems and finding a good solution constitutes very demanding work. Verifying the solutions and implementation are also big challenges. SDN architecture can significantly reduce this challenge by providing a single controller to apply all the solutions to and verify the solutions from. This is because in SDN architecture, the controller has a network-wide view that facilitates the formulation of the problems and the finding of solutions. From the polling the controller makes on the network elements, it has the desired knowledge about traffic load and resource constraints. A illustration of this idea is provided in Section 5.2.

TE can benefit from the programmability of SDN's centralized controller that facilitates automation and adaptability. The very complicated performance evaluation in practical networks can be done also in a centralized manner as the network elements report to the controller. This makes reaction times to problems in the network faster. In public IP networks, increasing the efficiency of network utilization, while minimizing the possibility of congestion, is a major challenge [3]. It is desirable for operators to reduce congestion to prevent delay and packet loss in order to facilitate a better user experience. Therefore, operators need to invest in TE in order to employ mechanisms that minimize congestion economically because in extreme cases congestion may collapse the network. CAPEX resources demand due to data growth, constitutes the single largest issue for network operators today and it will remain for a long term [30]. As such, it is a paramount importance to manage and utilize these resources efficiently.

As packet loss and delay are experienced when the load of the network is close to 100%, while approaching 70% or 80% load, a way of increasing the bandwidth of the link needs to be assessed and implemented. TE enables operators to keep utilization as low as possible for as long as possible without CAPEX, thus increasing profitability. When congestion is reduced, the QoE of users is improved, by avoiding wasteful retransmission. However, if no TE mechanism is implemented, it is not possible to know how the network reacts, thus coming up with good business model proves challenging for operators and SLA engagements and regulations will not be met.

4.5 Chapter Summary

TE is the endeavour by the network operator to facilitate the efficient use of bandwidth in the network in. Network operators need to do TE in order to realize the performance they have promised in the SLAs. In SDN architecture, the centralized

controller polls the network devices making the work of capacity management easier. For example, congestion management decisions such as routing traffic towards a different route as shown by the example in 5.2 happen faster. Moreover, it can inform end systems to decrease the rate they send traffic at. In traditional networking, expensive measurement systems to monitor traffic and report to TE mechanisms or tools are needed so that they take control action [3]. The capability of avoiding congestion and differentiation based on traffic characterization are core to modern TE. In the next chapter, congestion control application will be presented that shows SDN's capability of TE in MBH.

5 SDN Congestion Control Application (SCCA)

This experiment is done to show the feasibility of SDN for TE in MBH. It was showcased at the Mobile World Conference 2013 as part of Tellabs' proof-of-concept. The experiment is undertaken to show SDN based architecture doing congestion control. In this thesis, device statistics polling with a given parameter is used to obtain numbers, which served as a basis in the evaluation of SDN's scalability later in the analysis. The SCCA is SDN congestion control application prototype implemented in the Tellabs 8000 INM for Tellabs 8609 smart routers. When there is an increased utilization in the aggregation network, the congestion control mechanism redirects the traffic to a different path to avoid congestion. This section first expounds on the elements that constitute the testbed. Then, the testbed's layout is depicted and the implementation of the demo is explained. Finally, the demo use case is illustrated followed by different outcomes of the experiment displayed on graphs.

5.1 SCCA Testbed

This testbed is a scaled down representation of a MBH network. The devices used in this application are 8000 Intelligent Network Manager (INM), 6 Tellabs 8609 Smart Routers and Ixia N2X traffic generator.

5.1.1 INM

INM is explained in detail as it is used as a controller for this project. INM is a single and powerful network service management system that is supported on virtualized servers supporting Tellabs' MBH, smart routers and other solutions. INM does node troubleshooting and end-to-end service management across the entire network for 2G, 3G and 4G/LTE transport applications and services. In addition, it can also be used for pre planning networks, configurations and services before they are implemented. Service providers can manage business broadband Internet access and corporate voice services using INM. An advanced Graphical User Interface (GUI) hides the complexity in INM making network management easier and user experience better. The GUI is based on hierarchical windows which represent network elements and objects graphically, in tree and list view formats [35]. [36] The INM aims to provisioning faster, using its GUI that simplifies network management and has a remote configuration possibility to decrease operational costs.

INM only manages Tellabs' network elements such as routers, optical switches and Ethernet switches. INM offers service providers support for monitoring, troubleshooting and making changes of services. The INM can run on a single computer for small deployments and it can also scale up to 30,000 nodes with up to 150 concurrent users operating the network. It automates complicated tasks and used in planning future technologies such as the transition towards LTE and LTE-Advanced. [37]

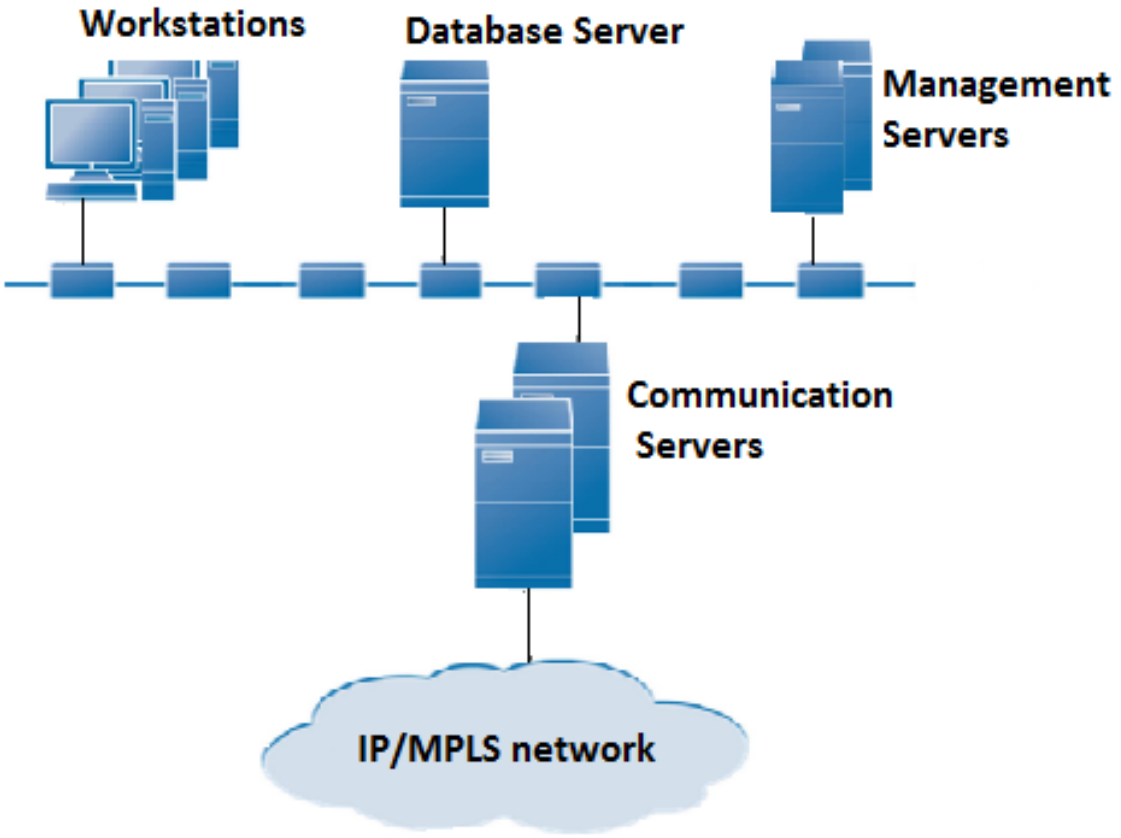

Figure 15: 8000 Intelligent Network Manager (INM) Configuration [35]

INM sits in a management LAN in the operator network which could be in a single or distributed physical locations depending on the number of nodes managed. It is separated from the WAN but can communicate with it through communication server. INM software uses 3-tier architecture that constitutes the GUI client, application logic (business logic) and data storage which are separated from each other and is possible to run on separate computers, for better scalability. Workstations are the computers the GUI is located in to access the INM. The business logic is where the tasks are done and it runs in a management server. In the management server, database communication and change notification are also performed. [35]

The communication server communicates with the WAN and manages the nodes. It sends configuration commands to the nodes, transfer files and polls the nodes for fault and performance information. This server maintains real time clock settings if the nodes do not support Network Time Protocol (NTP). The 8609 smart router, which is used in this thesis supports NTP. Nodes in a domain are accessed through adapters such as 8600 adapter in the communication server as shown in Figure 18. There can be one or more communication servers in a domain and they serve the areas assigned to them. Communication servers offer a fully redundant system by having backups during node communication. Management servers handle most of INM's business logic which is divided into service processes that can be run from

same or distributed servers. Workstations use the Satellite Service (gateway server) in order to use services of the Database Server, Management Servers and Communication Servers. [35]

The INM provides tools to simplify IP VPN provisioning which is used in LTE and LTE-Advanced networks. It has also tools to automate and simplify creation, configuration, reconfiguration and testing of connections. It enables the provisioning and management of the Virtual Private LAN Services (VPLS) infrastructure. Generally, troubleshooting and bringing about solution is facilitated by real time provisioning of IP VPNs, pseudowires, TDM circuits and VLAN VPNs assigned to a customer. Different reports are created graphically by INM from collected data about link utilization and traffic trends that helps to proactively provision bandwidth and avoid congestion. INM has tools for different tests of packet networks that could be performed from the GUI such as ping, trace route, delay, delay variation, throughput and packet loss. The database server is comprised of Sybase relational database where all INM tools store their data. Configuration data of managed networks are stored in the database. [37]

There are different packages in INM. The basic packages include network editor, fault management system, security management, OAM management and IP address management packages. The security management is comprised of operators access control, privileges to one or more part of the network, forming operator profile and Lightweight Directory Access Protocol (LDAP) and Remote Authentication Dial In User Service (RADIUS) authentication from a central server. [37]

5.1.2 Tellabs 8600

Tellabs 8600 Smart Routers are used from small aggregation sites to gateway sites. They extend MPLS-based services from the core network into access networks. They are designed to be LTE-ready and provide an extensive Ethernet and IP/MPLS feature set. They are backhaul platform that support various network technologies from different generations. Technologies supported include MPLS, RSVP, DiffServ, layer 3(IP) VPN, Layer 2 VPN and TE. They can be coupled with the INM for ease of management. They support different 2G, 3G and 4G technologies simultaneously. These smart router series product family is designed for providing managed 2G/3G mobile transport, Ethernet, broadband service aggregation and IP VPN services. [38] These smart routers are claimed to have offered high availability and have built in synchronization features that help in the smooth transition towards LTE and LTE-Advanced whereas increasing the capacity of MBH and managing from small cell aggregation to hundreds of gigabits at gateway sites. They are also able to implement Self-Organizing Networks (SONs) in MBH in order to automate tasks. [39]

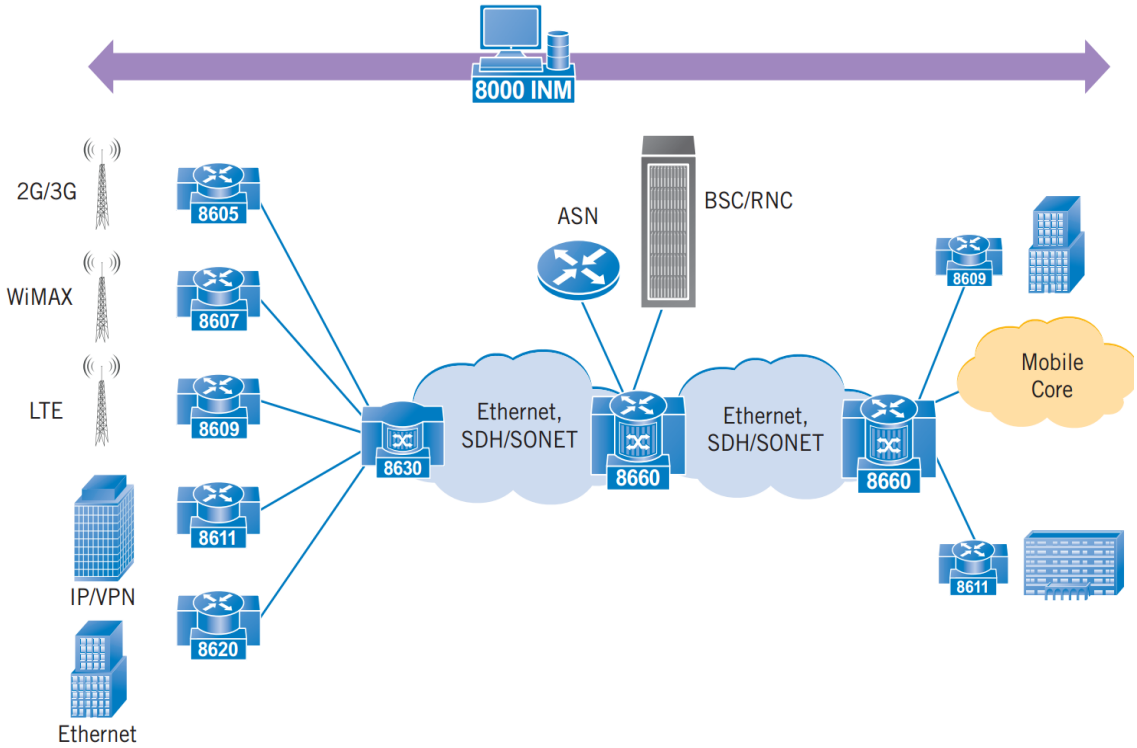

Figure 16: Tellabs 8600 routers support different generations of radio access networks and INM provides the management.

An example of IP/MPLS based aggregation site router is 8660 Smart Router, which had QoS awareness. It is a 14 rack unit (14 RU) router and supports different interfaces such as channelized TDM and Packet Over SONET/SDH (POS) to Ethernet and full redundancy. Its major applications are managed traffic aggregation in LTE, 3G and 2G mobile networks and delivery of Ethernet and IP VPN services. Its bi-directional switching capacity is up to 240 Gbps. [40]

5.1.3 8609 Smart Router

Tellabs 8609 Smart Router is a 1 RU high access router. It is a cell site optimized router which is optimal for aggregation as well. It has interface capacity for LTE and multi-protocol support providing 12 fixed GE ports and two slots for interface modules which offers throughput up to 7.5 Gbps. It can convert an E1/T1 TDM network to an Ethernet-based packet network without updating the access network equipment. This router is good for cell sites and access networks that serve ample amounts of mobile traffic. Its switching capacity is 5.5 Gbps and has multiservice E1/T1, FE and GE interfaces. QoS is performed for network optimization of voice and data services by this router's packet based forwarding architecture. It has similar feature of packet loop test as 8000 INM in order to evaluate performance. These tests are delay, jitter, throughput and other connectivity parameters. [41]

5.1.4 Layout of SCCA Testbed

The testbed consists of six Tellabs 8609 routers connected as shown in the figure below. Traffic is generated by an Ixia N2X traffic generator that is connected by single mode fiber optic cables to the cell site A, cell site B, Mobility Management Entity (MME) site and Serving Gateway (S-GW) site. The MME, the S-GW and the base stations are emulated inside the traffic generator.

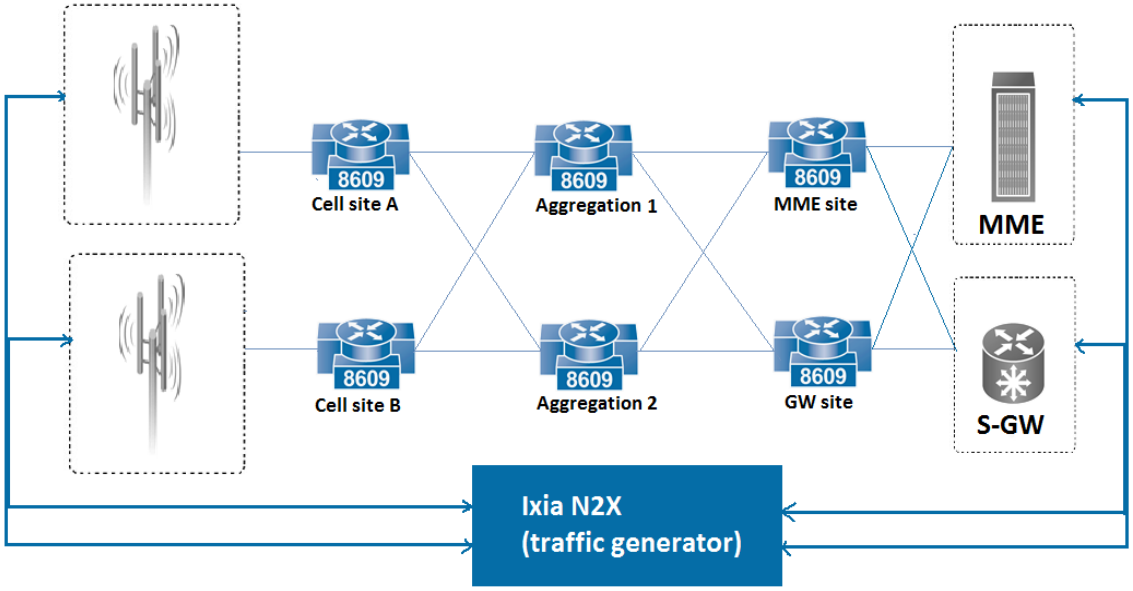

Figure 17: Logical layout of the testbed.

5.2 The Implementation SCCA demo

The controller has end-to-end network view and uses the OpenFlow protocol as a programming interface for this SDN architecture. The OpenFlow Protocol is a communication protocol between the controller and the data plane devices. OpenFlow is enabled in the communication server of the INM and the SDN controller is an add-on module on top of the INM's Business Logic Component (BLC) as shown in Figure 18. Which means, the main logic of the SDN application is in the BLC. In the 8600 adapter, there is a small piece of program that compiles OpenFlow structure. The 8609 routers monitor local link utilization and they periodically report this information to the controller.

The cell site routers are enabled by software to parse OpenFlow messages sent from the INM. The SDN controller knows about the whole network activity by polling on the devices. The INM profiled the threshold of the traffic load into high utilization and low utilization to determine the limits used for switching traffic to another path in this experiment. OpenFlow messages are sent to the cell site routers when switching is necessary. 10 seconds polling interval is chosen for this particular

implementation as it is assumed a good value for the demo; as polling too often increases overhead in the controller and polling too rarely would make the reaction speed slow.

The shared link in this application is the link between Aggregation 1 router and S-GW site as shown in Figure 20. In this experiment single peak values do not create reaction which means in order to prevent oscillations in the outcome, the algorithm is made to not react to single peaks exceeding or falling behind the threshold. Instead requires N consecutive samples above the limit to take action.

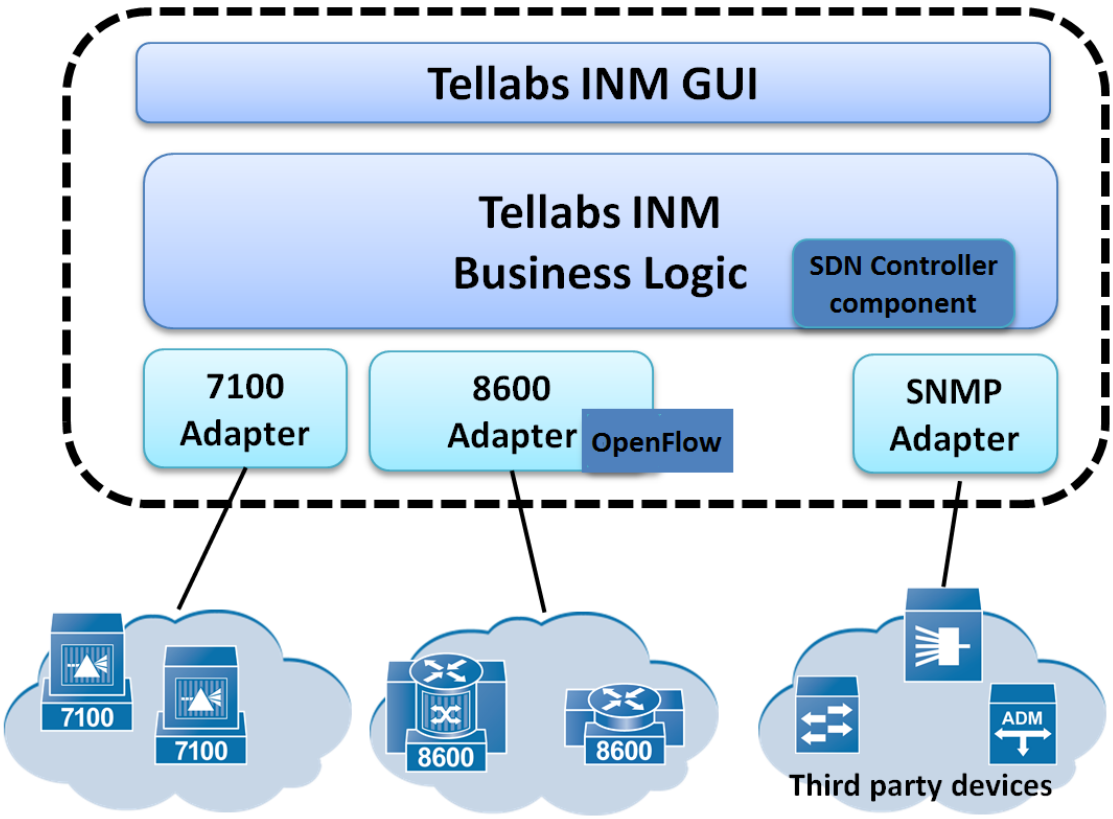

Figure 18: SCCA architecture.

5.2.1 Implemented Algorithm

The INM (Controller for this experiment) configures the tunnels using PWE3 and RSVP in advance and monitors traffic level on the links. It sends OpenFlow messages over TCP to the network elements (NE) with flow ID (to where the traffic is forwarded to). OpenFlow element in the NE receives the OpenFlow message. In the NE, database mapping is done between interface and pseudowire during basic configuration when pseudowire is attached to interfaces. Pseudowire is attached from the ingress on cell sites to egress of S-GW site and RSVP tunnel from the egress of cell sites to ingress of S-GW site. Flow ID matches to the affinity of the pseudowire. The OpenFlow configuration agent in the NE configures the flow id to hardware.

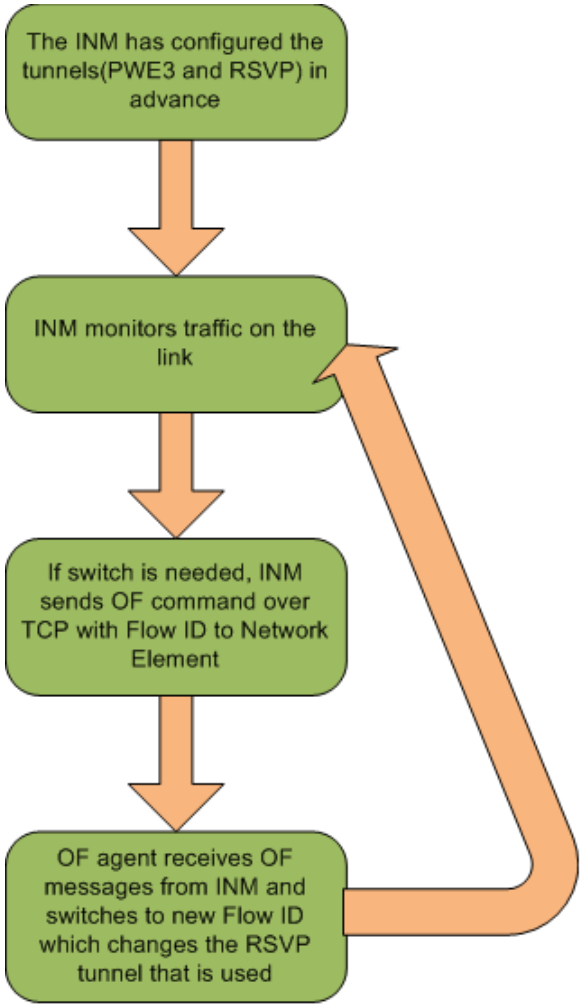

Figure 19: The workflow.

5.3 SCCA demo

In this demo, the traffic generator was generating Ethernet traffic with IPv4 in the payload of Ethernet through the cell site routers. Let's call the traffic from cell site A to aggregation 1, traffic A and the traffic from cell site B to aggregation 1, traffic B. The solid red and the blue lines both routing through aggregation 1, show the optimal traffic path for moderate traffic scenario. The SDN controller polls on the devices every 10 seconds. The INM has set high utilization threshold of 60 percent and low utilization threshold of 20 percent for traffic load on the shared link. If this high threshold is exceeded, high utilization is observed and traffic B is switched to pass through aggregation router 2 as shown by the yellow line in Figure 21. This is achieved by SDN controller sending OpenFlow messages to the cell site routers and switches the traffic B to the unutilized link to avoid congestion.

In the MWC Tellabs demo depicted in Figure 20, the blue flow illustrates traffic from residential areas as an example and the red flow represents traffic from a shopping mall. In the demo, it is assumed the optimal traffic path is through aggregation 1 for both traffic sources as long as there is a moderate traffic volume. INM performance monitoring window shows the link utilization based on the ingress and egress traffic of Aggregation 1 as shown for example in Figure 22. We can increase and decrease traffic from the control panel of the traffic generator's control panel. The VPN provisioning window shows the current route of the traffic as shown in Figure 23.

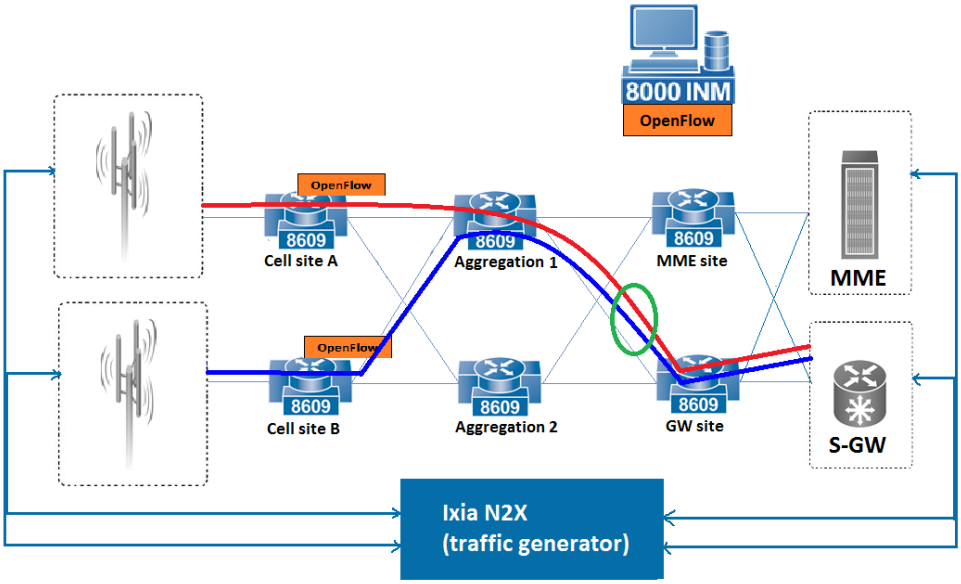

Figure 20: Optimal traffic pattern.

In late afternoon, as the traffic from the shopping mall increases heavily and the link utilization goes up in the shared link causing it to go above 60 percent, the controller sends OpenFlow messages to the cell sites. The blue flow from the residential area is assumed to be lower priority traffic and is directed to the less optimal route through aggregation 2 as shown by the yellow line in Figure 21. From the INM VPN provisioning window we can now see the switch to the new traffic path as shown in Figure 25. The switching makes the link utilization of the shared link lower and prevents congestion for the higher priority traffic from the shopping mall.

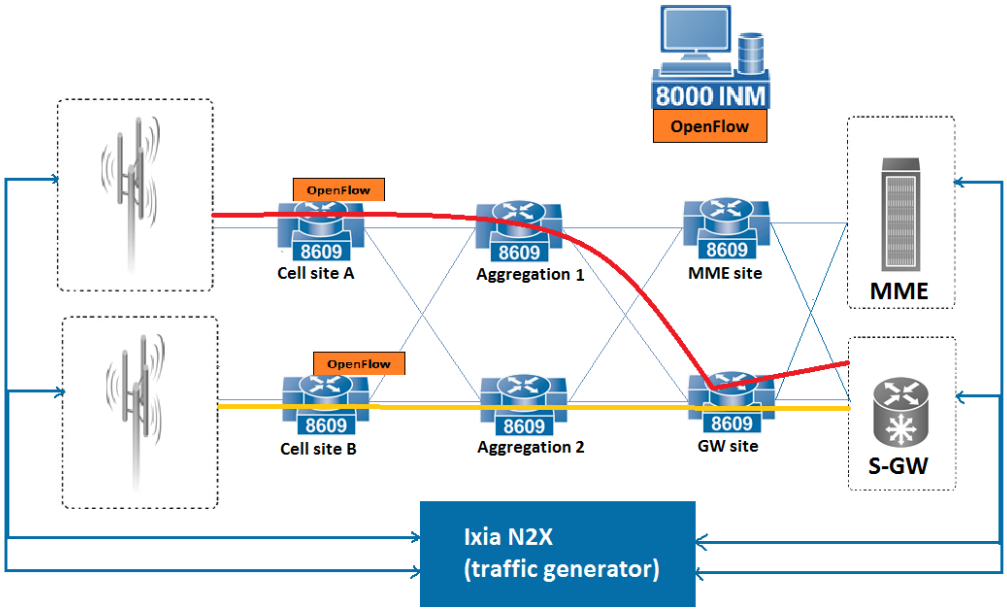

Figure 21: High utilization traffic pattern.

The following graphs showcase utilization per time in different scenarios. Traffic A is depicted on the top left, traffic B, on the bottom left and the shared traffic on aggregation 1, on the right portion of the graph.

In the case where A is set to 15% and B is set to 25%, the shared link shows roughly their sum which is about 40% which is the traffic utilization on the ingress of the shared link. This is still under the high threshold therefore, high link utilization is not observed and the shared link handles traffic from both sites as shown in Figure 22.

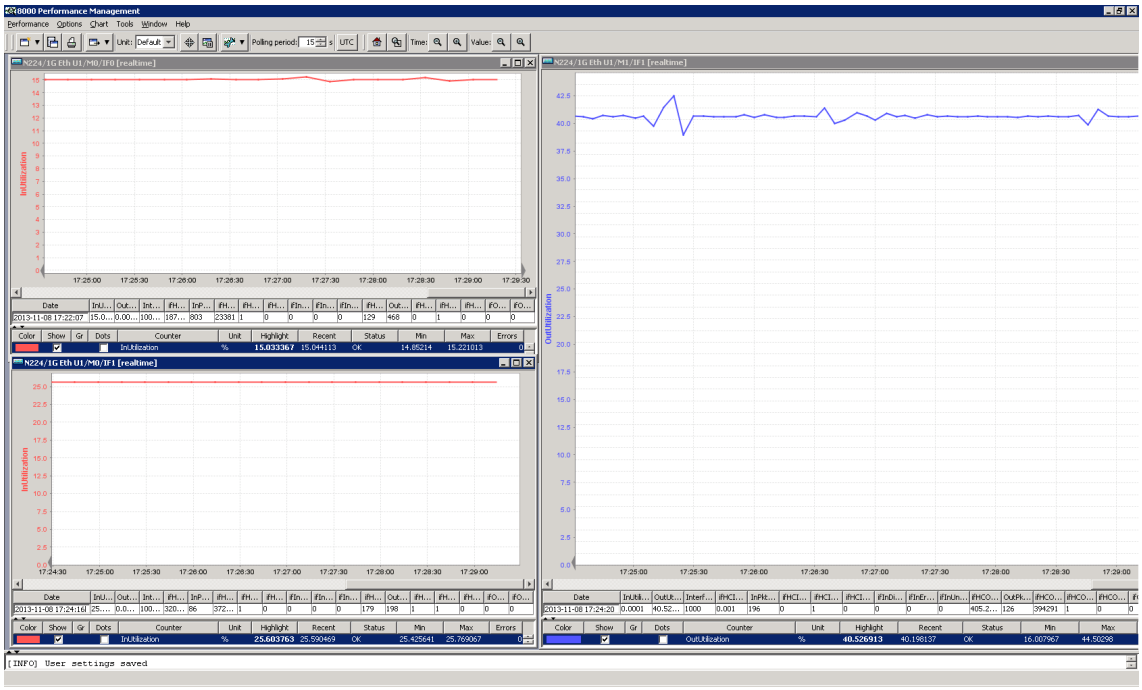

Figure 22: Moderate utilization of the link.

The INM VPN provisioning window shows B is through aggregation 1 as shown in Figure 23. If there is no traffic from A, the shared link shows traffic from B as long as it is under the high threshold value.

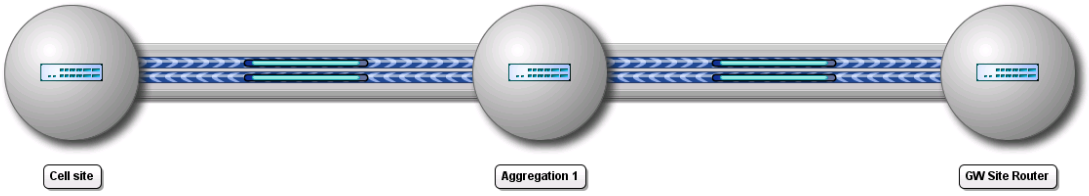

Figure 23: A and B using the shared link.

As A increases from 15% to 50%, for the duration of the polling (10 sec) interval, the shared link shows the sum of A and B which is 75%. After the polling is done and OpenFlow is activated, since a value greater than the high threshold is detected, the shared link now only shows 50% (only traffic A). B shows a drop from 25% to 0 on the shared link because it is switched to go through Aggregation 2 and aggregation 1 ingress does not see B. This is depicted in Figure 24

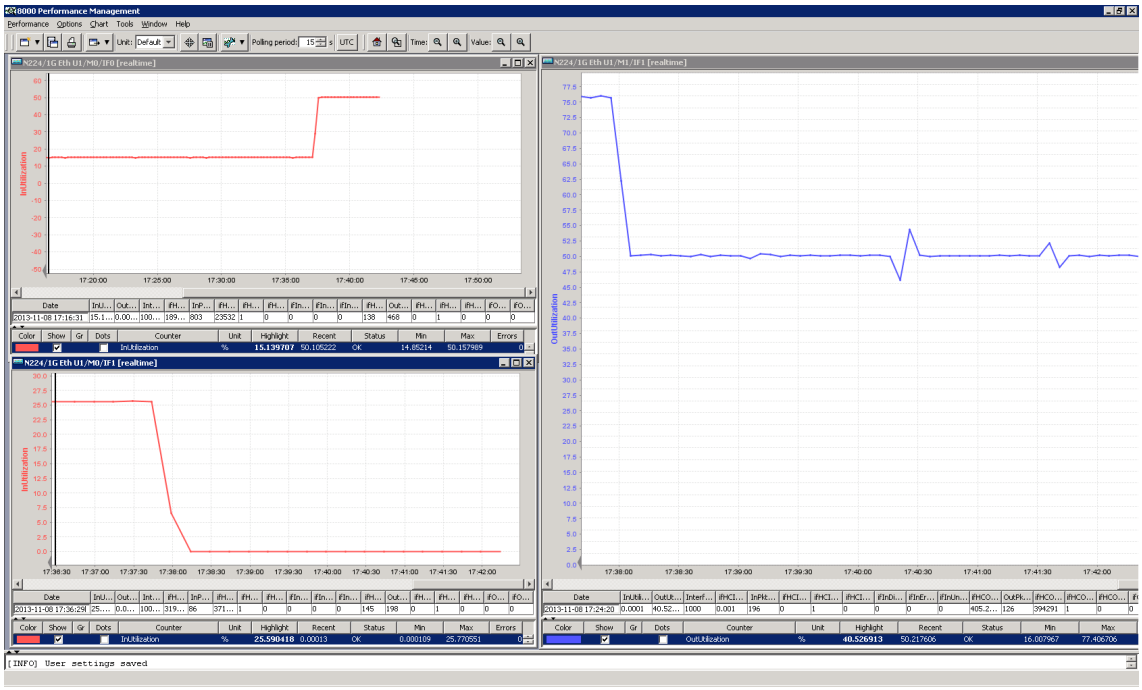

Figure 24: High utilization detected in the shared link.

The VPN provisioning window depicts this concept by adding aggregation 2 for B as shown in Figure 23.

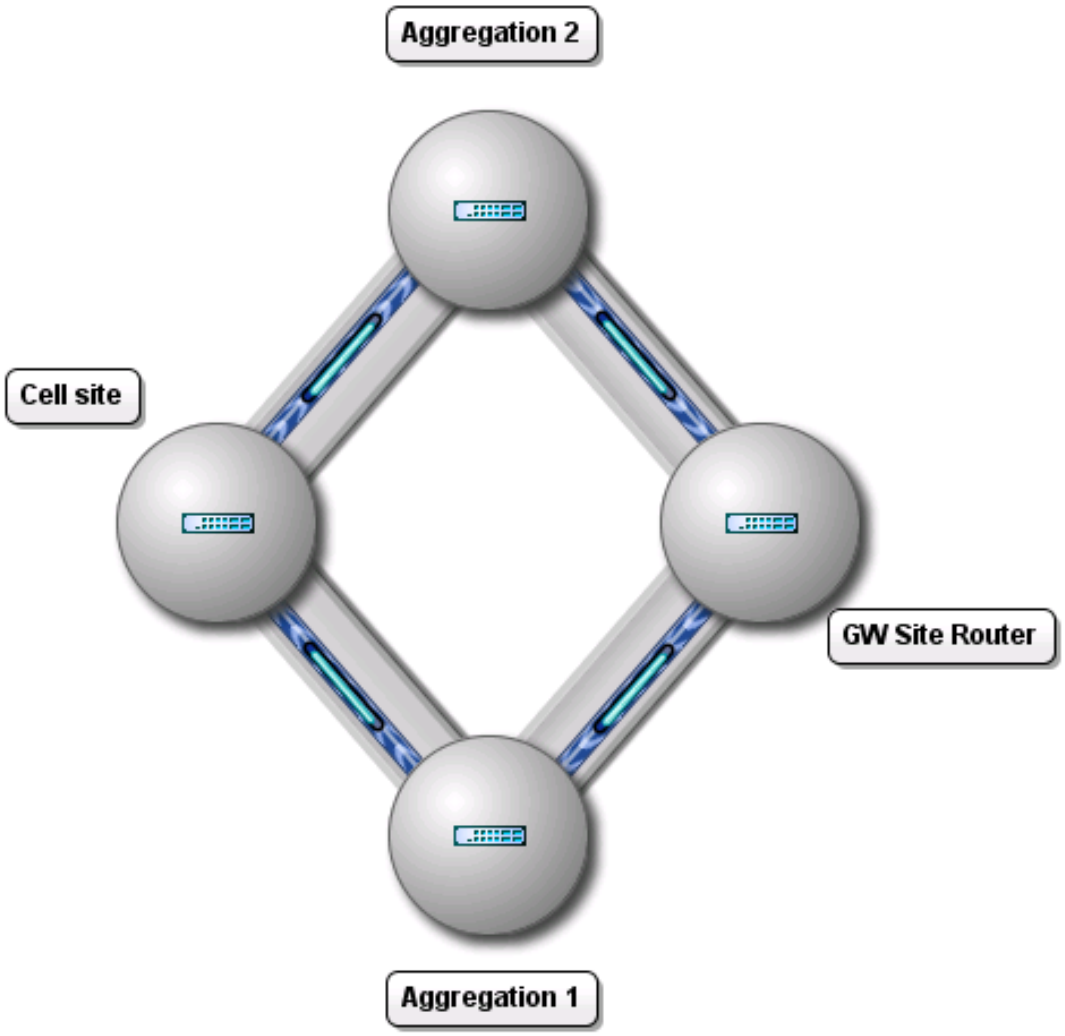

Figure 25: B switched to use the less optimal path.

When A is decreased from 50% to 25% and B remains at 25%, the shared link only serves A as shown in Figure 26 even if the sum is under the 60% threshold. This is because the low threshold of 20% is not detected in the shared link and the system does not switch B back to aggregation 1. Therefore the VPN provisioning window still shows the diamond shaped link as shown in Figure 25 above.

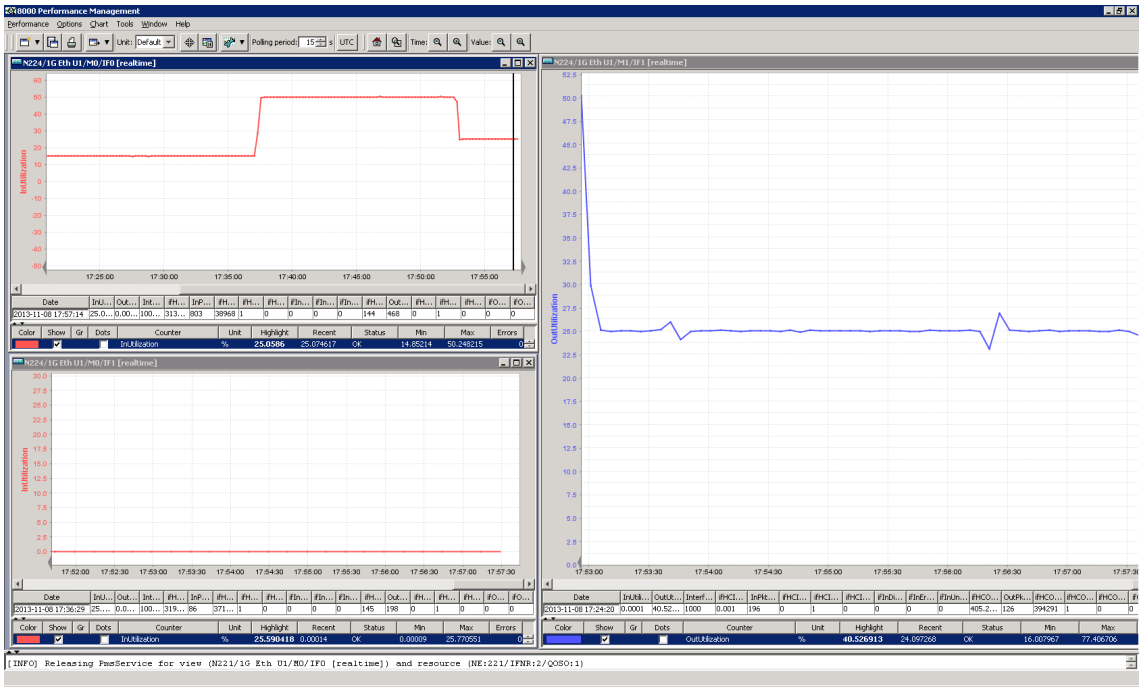

Figure 26: B continues to use the less optimal path.

When A is decreased to 15% from 25% while B remains at 25%, a value lower than the low threshold(20%) is detected and B is switched back to go through aggregation 1, after polling is completed as shown in Figure 27. Aggregation 1 now shows the sum of A and B which is similar to Figure 22 after polling. The VPN provisioning window shows a straight line through aggregation 1 (Figure 23).

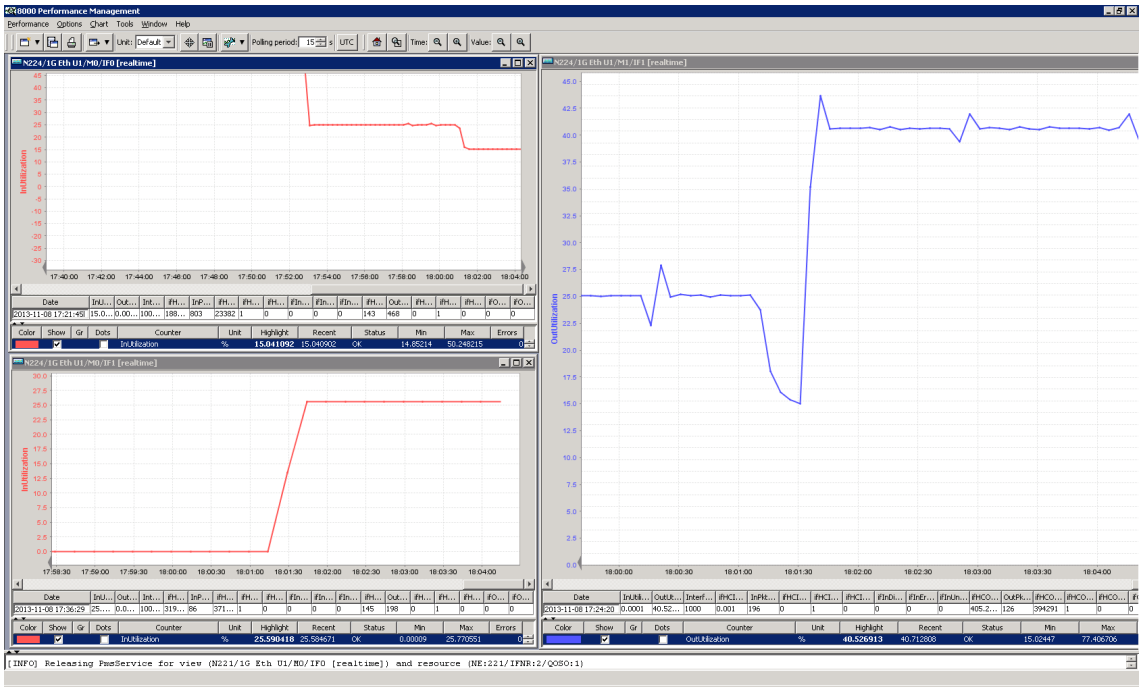

Figure 27: Low utilization detected in the shared link.

The case where B is set to increase to 60% while A is still held at 15% causes fluctuation. First the polling shows there is high link utilization of 75% and switches B to pass through aggregation 2 but then it detects low link utilization and switches B back to the shared link and this goes on until A is set to increase over 20% or B is set to be less than 45%. The shared ingress will have traffic B and then after next polling it will only have traffic A and this goes on until traffic conditions change, as shown on the bottom left of the performance management window in Figure 28.

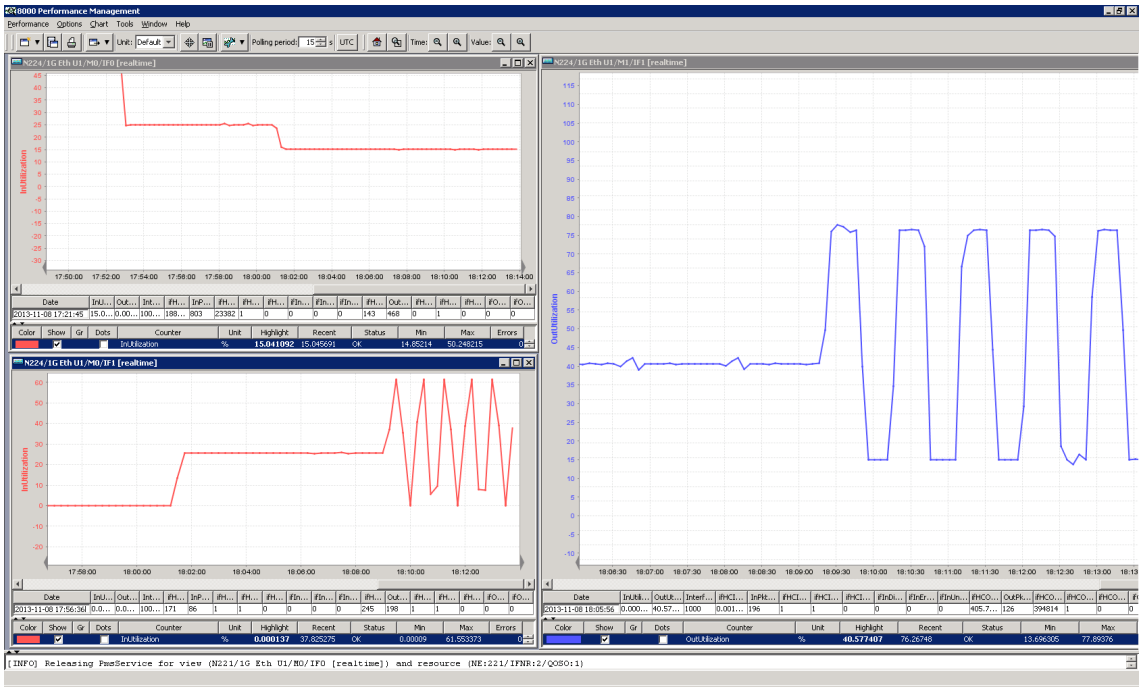

Figure 28: Oscillation between high and low utilization.

The VPN provisioning window shows the links altering from diamond to straight shape and vice versa with each change following the fluctuation. This has to be improved in future development. Note that if A is set to over 20% and B was 60%, unless A goes under 20 the aggregate traffic will only have A even if B is decreased so that the sum is under 60. The traffic from A is the priority and the decisive traffic.

5.4 Chapter Summary

The congestion control application illustrated in this chapter is a proof-of-concept product to showcase SDN in action. It brings together ideas from all previous chapters as such it represents a MBH network and employs SDN with OF for the purpose of TE. It has met its goal of showcasing load balancing solution using SDN. It was possible to show outcomes using different graphs and figures. Moreover, as it is performed on real networking devices the feasibility of introducing SDN is realized in this case for small scale networks. It took into consideration a meaningful and useful scenario of TE that is time of day based traffic pattern. It is apparent that it has limitations by the oscillation depicted in Figure 28. In the analysis section this will be further discussed.

6 Analysis and Discussion

This section analyses the application made in chapter 5, its use cases and its limitations. Moreover, other SDN related interesting insights will be discussed.

6.1 Analysis of the SCCA

The SDN Congestion Control Application (SCCA) showcases SDN in action for TE in MBH. It is a demonstration of SDN feasibility on commercial off-the-shelf hardware, depicting a network based on user traffic patterns. The results are displayed on a time series graph. It is a load balancing solution example that can be carried out to counteract congestion. Traffic analysis is done by the controller on the shared link. This demonstrator has limitations among which : it consists of only six nodes, there were a limited number of traffic sources and types and it exhibits the problem of oscillation. The limitations will be discussed in detail in this section.

For future research this experiment needs to be done on a much bigger scale to incorporate global optimization algorithms considering constraints imposed by many nodes and conflicting interests. That is, congestion solutions for one section of the network may conflict with other solutions from a different section. Moreover, in real networks there are many traffic sources and optimization that is all inclusive is essential. The algorithm will not be simple as having only two paths; therefore it needs to be done carefully. Collecting empirical measurement results from big enough number of nodes, will help to properly represent real network sizes through extrapolation. It will help to estimate the real overhead that is caused by polling and reporting. Therefore, for the future, this experiment needs to be done with many nodes or in simulation studies of appropriate scale.

Averaging was used to avoid oscillation in this experiment but as shown in Figure 28, it was not enough to avoid fluctuation. Oscillation is not desirable in this network due to the unnecessary signalling overhead across the network and additional computation it imposes on the controller. A stronger mechanism is needed to prevent this. In future developments of this project, one possible approach could be a threshold mechanism using hysteresis. Hysteresis characterizes a system that has output which does not depend only on its current input but also on its past behaviour on the path it has followed. It is used to avoid unwanted switching due to fluctuations.

One way of applying hysteresis is introducing a phase lag between the input and the output to make the output react slower. When the traffic A is less than the minimum threshold, the controller could check traffic B to calculate the total traffic amount. If the traffic B added with traffic A causes high utilization, it will keep routing B through aggregation 2 which is the "less optimal path". During the time the mechanism is aware and starts functioning the hysteresis will lag the output and reduces the unwanted switching. Hysteresis need to be moderated because if bigger

hysteresis margin is used the optimal route might be used less often than necessary, wasting bandwidth.

6.2 SDN's Controller Scalability

Scalability represents a major concern for future SDN deployments. Due to a central controller, questions may arise as to whether the controller is scalable to support real networks with many nodes. One such area of concern is the overhead generated by polling network elements. This procedure involves the controller sending an OpenFlow request message to every device under its supervision in order to obtain statistics from the device. These statistics are used as inputs to the traffic control algorithm.

A sample calculation is provided below to estimate the OpenFlow messaging load from network devices to their controller. It is assumed each device reports every 10 seconds statistics on flows on an aggregate level (OFPMP_AGGREGATE), flow tables (OFPMP_TABLE), ports (OFPMP_PORT_STATS) and queues for a port (OFPMP_QUEUE). Each of these constituted of an OFPT_MULTIPART_REPLY and one of the type-specific fields listed in Table below. The former includes the OpenFlow header(8 bytes) along with additional fields that total 8 bytes. The OpenFlow header is at the beginning of all OpenFlow messages.

Table 6: Message Types used.

Message Type	size
OFPMP_AGGREGATE message	24
OFPMP_TABLE	24
OFPMP_PORT request reply	112
OFPMP_QUEUE message reply	40

There are different message types and the choice of which to use is based on the operation being performed. When switches respond to the controller's polling for table state, they use one or more OFPT_MULTIPART_REPLY messages. The exact number depends on how many different types are requested. The type in OFPT_MULTIPART_REPLY field indicates, the kind of information being passed and resolves what to do with the body field.

One device reporting Aggregate flow statistics will have : (OpenFlow header + OFPT_MULTIPART_REPLY (with out OpenFlow header) + OFPMP_AGGREGATE message)

$8 + 8 + 24 = 40$ bytes

Assuming there are 200 flow tables : (OpenFlow header + OFPT_MULTIPART_REPLY (with out OpenFlow header) + OFPMP_PORT request reply * number of tables)

8 + 8 + 24*200 = 4816 bytes

Assuming each device to have 4 ports : (OpenFlow header + OFPT_MULTIPART_REPLY (with out OpenFlow header) + OFPMP_QUEUE message reply * number of ports)

8 + 8 + 112*4 = 464 bytes

This queue statistics reply could be for more than one port and queue. If we assume two queue statistics per port and calculate the replied amount of bytes :

8 + 8 + 40*8 = 336 bytes

Adding up all the constituent parts of the network element's reply yields:

40 + 4816 + 464 + 336 = 5656 bytes

Assuming these data is handed to TCP in a single transfer and further assuming a 1500 byte Ethernet Maximum Transmission Unit (MTU), 20 bytes IPv4 header and 20 bytes TCP header, results in the following division into packets:

5656 + 40 - 1500 = 4156
4156 + 40 - 1500 = 2696
2696 + 40 - 1500 = 1236
1236 + 40 = 1276

Therefore, each OpenFlow statistics report message will be segmented into 4 IP packets. The total reporting message per device, per report with the assumptions taken for this analysis, is 5656 bytes of OpenFlow message + 160 bytes IP and TCP headers that results in 5816 bytes. Since the controller polls every 10 seconds, there will be 581.6 bytes per second reported per device on average. If the number of nodes is 10000, 581.6 bytes * 10,000 = 5,816,000 bytes/second are sent to the controller. That is 5.55 MB/second or 44.4 Mbits/second. Commodity GE links can easily carry this traffic.

The processing power of current powerful servers is sufficient to handle hundreds of thousands packets/second even without optimization applied and up to 80 million packets/second with optimization [42]. With the above assumptions this results in 4000 packets/second being generated. This leaves significant processing capacity margin in the controller to handle other tasks, such as traffic engineering optimization in its network.

With N = 10,000 devices and polling every t = 10 seconds, there will be 1000 re-

Table 7: Software and hardware specification of comparison NOX controller controller.

CPU	8 cores at 2.0GHz
RAM	4GB DDR2
Network	2x1000Mbit/sec bonded
OS	2.6.32-5-686-bigmem
Compiler	GCC 4.4.5
libc	libc 2.11
libC++	GNU lib C++ 3

quests/second for the controller. With t = 1 second, it will be 10,000 requests/second. A single threaded NOX controller without optimization for performance can handle 30,000 requests/second [43]. This number proves feasibility as most controllers are being developed with optimizations applied for performance and have many folds improved performance compare to NOX. Moreover, often not all switches are in an active state simultaneously in real networks resulting in each controller being able to manage large number of switches [43].

The size of the controller database and the interaction between controllers may pose a challenge. A distributed controller architecture has been proposed in many studies to share the burden of a single controller. [44] This would however increase the signalling overhead as all controllers must maintain a coherent view of the network. In light of the above calculation, this would seem unlikely to constitute a major hurdle.

Alternatively or in conjunction, the network can assign some tasks from the controller to other dedicated devices to do certain off load tasks in order to alleviate the controller's burden. This approach has been successfully used in other fields. For example, database servers running on a separate server from the one running the http server of a website. Instead of a separate server, this off-load device could also be an accelerator module supplementing the CPU. This enables a massive speed up of computation due to the specialized nature of the former. Another option is the use of intermediate local aggregation controllers to reduce the control traffic towards the main controller. These local controllers will be responsible for implementing the main controller's commands in a small area.

6.3 The Transition to SDN

Mobile operators these days face major challenges with handling the massive volume of mobile data traffic. Therefore, they need to increase the capacity of their network. We have seen that efforts have been made to tackle this problem such as deploy-

ment of small cells. The different efforts undertook are resulting in heterogeneous networks with increased complexity to manage and provision. The management of these networks is becoming expensive and operators are constantly looking for ways to make their network profitable; the raised management complexity is making the centralized controller an appealing feature to have.

ASICs exhibit low flexibility but could have speed up to 1000 Gbps. They are made for a specific purpose and upgrading for a new service is very challenging. The multiprocessors used in general purpose architectures, however, have speeds only up to few tens of Gbps even after an optimization such as load balancing between different cores is performed. This speed is not comparable to the custom ASICs. [44] It is interesting to discuss how a hybrid of these two strengths can be deployed. Specific heavy tasks that are an integral part of the operator's network can be performed by ASIC-based devices while the more flexible elements can be deployed in the rest of the network for other tasks. It is apparent in this situation, SDN is not in its purest form but this approach can be the center of interoperability with - and transition to - SDN.

The SDN architecture - as has been mentioned - makes the switches to be managed very simple creating the possibility of cheaper hardware. This decreases the CAPEX for service providers. The management being done from a single or few controllers, most problems that arise are consolidated and hence faster to troubleshoot and solve, resulting in decreased OPEX. The classical and hybrid approaches to SDN are available. The hybrid approach makes the transition to SDN easier. Interoperability with existing legacy networks need to be dealt with because there are many such systems supported by these networks. Currently due to the existing expensive legacy architectures still in use, SDN networks need to be deployed incrementally by being introduced in the legacy networks.

SDN is beneficial in MBH. MBH accounts for about 40% of the total network related mobile operators cost. The heterogeneity of the telecommunication networks presents great challenges to managing them. Different organizations have their own management systems that are quite expensive to maintain. These systems are often proprietary and do not play parts in simplifying the devices being managed. They often can only be used by the organization or customers of the organization. When managing heterogeneous networks the network management systems suffer from huge complication. SDN on the other hand simplifies the underlying network thus simplifying the assignment and management of resources.

Once SDN-type devices have achieved sufficient penetration instead the network, older legacy management systems ca start to be decommised gradually. When all have been removed, the transition to an SDN architecture is essentially completed. SDN architecture has a controller that orchestrates the network fabric. SDN backhaul solution is easy to install and run and that will save the operators deployment cost and time. Intelligent network devices are not necessary hence less expensive

routers that now only do mere forwarding, while the decisions are made in a separate controller. The controller uses software to make decisions about the forwarding plane. It can see the network and can make centralized path calculation and direct the traffic in a desired path. This increase the amount of control operator has over the network. This in turn simplifies the OAM and creates a fast time to market for services. OAM system is very expensive simplifying it has cost advantages.

6.4 SDN for SLA

Operators need to fulfil SLA parameters to meet QoS for their customers. Some very important SLA parameters for networks applications and services are threshold in delay, delay variation, packet loss rate, throughput due to congestion, availability, and per flow sequence preservation [45]. These parameters govern how the service or the application is experienced and as the operator has contracts with customers to give the qualities mentioned in the SLA, they need to be meticulously managed. Throughput often is determined by the slowest link. Throughput needs to be enhanced to support the increased demand for traffic. The SDN controller having full network view can help throughput by - for example - choosing the longer route with more bandwidth rather than shortest path. The MBH's transport network needs to be improved to fulfil throughput requirements. SDN is a crucial enabler of a new infrastructure. SDN controller can select paths and assign resources while caution can be taken to fulfil the SLA requirements by the service, all from a centralized means.

6.5 Realtime Fault Recovery

Bidirectional Forwarding Detection (BFD) is a protocol that detects faults fast in bidirectional path between two forwarding devices connected by a link. [46]. Multiple BFD sessions need to be established if there are multiple connections between two routers. In realtime fault recovery by BFD, if a device does not get any BFD packets with a given detection time limit, it assumes the link is broken. MPLS-TP is required to provide mechanisms that guarantee 50ms recovery times (if the network is within 1200 km) that is calculated from the moment the fault is detected [47]. In SDN that includes fault notification time, controller processing time, flow modification to correct it and distribution to all affected nodes. A study found that convergence time of an unknown flow is 50ms or less for about 70% of flows to reach their destination in SDN network. This means 30% will exceed the MPLS-TP upper bound for recovery time [48]. In SDN controller the backup route could be pre-calculated and the device may switch to it upon fault detection. The network element can report what happened to the controller later. Depending on the application, different code optimization could be done to make it react faster. This however shows SDN and OpenFlow need to consider the realtime issues in the future specifications.

6.6 Chapter Summary

A protocol and clear standardization need to developed in order to incorporate SDN functionalities and provide backward compatibility to work with existing technologies. [44] It has been discussed that the controller is not a bottleneck based on statistics reporting load. As other studies have suggested the bottleneck likely lies elsewhere. However, as there have not been enough studies on the performance of the controller so far, it calls for different dimensional studies in this area. Should it prove to be a limiting factor for performance, several approaches exist to reduce the dependency on one device for all computation. Additional controllers or accelerators can reduce the load on the main controller while aggregating multiple nodes behind one local controller reduces the number of managed elements for the controller.

7 Conclusion

One critical challenge faced in device design is that an increase in programmability generally means a corresponding decrease in performance of a processor. In SDN, more flexible and programmable general purpose processors are used. These processors have reduced speed custom made ASICs, however, they exhibit high flexibility due to programmability. ASIC-based devices have very high performance but they are time consuming to make and expensive. To tackle this trade-off between programmability and performance, a hybrid of SDN and legacy systems approach could be used in order to meet the performance requirements.

SDN is a new paradigm and a large amount of research is ongoing. It is necessary to do these researches before wide scale deployments. In this thesis, the scalability issue is discussed and analysed. However for the future performance of SDN controller needs to be studied further. As shown in the analysis, device reporting overhead has been shown to not constitute a bottleneck for the deployment of SDN networks. However, other aspects of the controller such as security and availability need to be researched and modelled more for the future. It is an integral and very important part of the SDN architecture and widespread research is needed before wide scale deployment. Figuring out the best algorithms to performing network wide optimization is essential.

With the evolution of cellular technologies, SDN will only gain more use cases. In LTE, all-IP network calls for an all-IP MBH with easier management of the network and more bandwidth flexibility however, the backhaul is not programmable enough even if it evolved to a more flexible technology. Accommodating the ever increasing volume of mobile data traffic, by the numerous applications being made for smart devices, proves difficult in an inflexible network. SDN architecture in the backhaul can revolutionize the development of the MBH by increasing its programmability. The devices in MBH can be simplified to mere forwarding elements managed by a remote controller.

Due to the fact that 2G, 3G and 4G networks are all simultaneously working and still in use in current networks, SDN is introduced within the existing mobile network rather than the pure SDN approach. SDN also helps in the interoperability of the different technologies by managing the interworking from a central controller. Vendors are including OpenFlow capability to their network devices currently which shows a future for SDN deployment. SDN significantly reduces vendor lock-in for operators as well as providing them with new opportunities for revenue generation and agile service creation.

References

[1] Cisco . Cisco Global Cloud Index: Forecast and Methodology, 2013 to 2018. *White Paper*. 2014. [cited 03 Nov 2014]. Available at:http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf.

[2] Tullberg Hugo, Li Zexian, Höglund Andreas, Fertl Peter, Gozalvez-Serrano David, Pawlak Krystian, Popovski Petar, Mange Geneviéve, Bulakci Ömer . Towards the METIS 5G Concept. 2014. [cited 03 Nov 2014]. Available at:https://www.metis2020.com/wp-content/uploads/publications/EuCNC_2014_Tullberg_etal_Towards-the-METIS-5G-Concept.pdf.

[3] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja and X. Xiao. Overview and principles of Internet traffic engineering. *Internet Engineering Task Force: RFC 3272*. May 2002. Also available at:http://tools.ietf.org/html/rfc3272.

[4] SDNCentral. Why SDN or NFV now. 2013. [cited 02 October 2014]. Available at: https://www.sdncentral.com/why-sdn-software-defined-networking-or-nfv-network-functions-virtualization-now/.

[5] SDR Forum. SDRF Cognitive Radio Definitions. 2007. [cited 2 April 2013]. Available at:http://www.sdrforum.org/pages/documentLibrary/documents/SDRF-06-R-0011-V1_0_0.pdf.

[6] Kimery J. Shearman, S. Software Defined Radio Prototyping Platforms Enable a Flexible Approach to Design . *Microwave Magazine, IEEE*, Vol.13:5. 2012. [cited 2 April 2013]. Available at:http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6238541.

[7] Open Networking Foundation. Software-defined networking:the new norm for networks. *ONF White Paper*. 2012. P. 2-12. [cited 25 March 2013]. Available at:https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

[8] Tellabs. SDN network architecture - Tellabs SDN demo MWC 2013.

[9] The Open Networking Foundation. OpenFlow Switch Specifcation Version 1.3.0 . June 25, 2012. [cited 25 April 2013]. Available at:https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf.

[10] Azodolmolky Siamak. *Software Defined Networking with OpenFlow*. Packt Publishing, 2013. 152 p. ISBN 978-1849698726.

[11] Juniper Networks. Universal Access and Aggregation Mobile Backhaul Design Guide 1.0. Nov 2013. [cited 2 Jan 2013]. Available at: http://www.slideshare.net/junipernetworks/universal-access-and-aggregation-mobile-backhaul-design-mobile-backhaul-design-guide.

[12] Inc. RCR Wireless. Mobile Backhaul Resource Guide . 2013.[cited 1 Mar 2013]. Available at:http://www.rcrwireless.com/mobile-backhaul/2013-mobile-backhaul-resource-guide.htmll.

[13] NEC Corporation. Mobile backhaul evolution and convergence. *WHITE PAPER. [E-SEMINAR].* Jan 2010.[cited 26 April 2013]. Available at:http://www.nec.com/en/global/ad/mbh/pdf/wp.pdf.

[14] Tellabs. Smart mobile backhaul for successful het nets. *WHITE PAPER.*, Vol. 22:11-12. P. 12 - 14. [cited 25 March 2013]. Available at:http://www.tellabs.com/products/8000/tlab8600sysoverview.pdf. issn 1532-2459.

[15] Ericsson. It all comes back to backhaul. 2012.[Cited 14 April 2013]. Available at:http://www.ericsson.com/res/docs/whitepapers/WP-Heterogeneous-Networks-Backhaul.pdf.

[16] Holma Harri, Toskala Antti . *LTE Advanced : 3GPP Solution for IMT-Advanced.* Wiley, 2012. 249 p. ISBN 9781118399415.

[17] Juniper Networks. Mobile backhaul reference architecture. *Reference Architecture.* 2011. [Cited 11 April 2013]. Available at:http://www.ictnetworks.com.au/pdf/8030008-en.pdf.

[18] W Goralski. *The Illustrated Network.* Morgan Kaufmann, 2008. 832 p. ISBN 9780123745415.

[19] Davies Guy. *Designing and Developing Scalable IP Networks.* John Wiley and Sons, Incorporated, 2004. 270 p. ISBN 0-470-86739-6.

[20] Cisco. MPLS Traffic Engineering and Enhancements . *Cisco IOS Software Releases 12.0 S.*

[21] E. Rosen, Y. Rekhter. BGP/MPLS IP Virtual Private Networks (VPNs). *Internet Engineering Task Force: RFC 4364.* February 2006. Also available at:http://tools.ietf.org/search/rfc4364.

[22] S. Bryant, Ed.Cisco Systems P. Pate, Ed.Overture Networks, Inc. Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture. *Internet Engineering Task Force: RFC 3985.* March 2005. Also vailable at:http://tools.ietf.org/search/rfc3985.

[23] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018. [White Paper] 2014.[cited 5 Feb 2014]. Available at:http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.

[24] Masha Zager. WIRED NETWORKS AND THE BACKHAUL BONUS. 2012. [cited 15 Jan 2014]. Available at:http://www.ruraltelecom.org/march/april-2012/wired-networks-and-the-backhaul-bonus.html.

[25] Salmelin Juha, Metsala Esa. *Mobile Backhaul.* Wiley, 2012. 410 p. ISBN 9781119941026.

[26] Zhao Fuchuan Cong Kai. Ptn and ip-based mobile backhaul. *ZTE communications Publications.* 2010. [cited 2 September 2014]. Available at:http://wwwen.zte.com.cn/endata/magazine/ztecommunications/2010Year/no3/articles/201009/t20100913_191837.html.

[27] SDNCentral. What is nfv network functions virtualization. 2013. [cited 02 October 2014]. Available at:https://www.sdncentral.com/whats-network-functions-virtualization-nfv.

[28] Harish Viswanathan Colin L. Kahn Mark M. Clougherty, Christopher A. White. The role of sdn in ip network evolutionl. *Paper.* 2014. [cited 12 October 2014]. Available at:http://www.telecomsci.com.cn/config/newsfiles/2014052091024001/sdn.pdfl.

[29] Lancy Lobo Umesh Lakshman. MPLS Traffic Engineering . Jan 13, 2006. [cited 1 Nov 2013]. Available at:http://www.ciscopress.com/articles/article.asp?p=426640.

[30] EYGM Limited. Future network operations. 2013. [cited 09 Sep 2014]. Available at http://www.ey.com/Publication/vwLUAssets/Future_network_operations/$FILE/Future_network_operations.pdf.

[31] Kenichi Mase Shigeo Shioda. Performance comparison between intserv-based and diffserv-based networks. *Global Telecommunications Conference Publication.* 2005. [cited 01 October 2014]. Available at:http://ieeexplore.ieee.org.libproxy.aalto.fi/stamp/stamp.jsp?tp=&arnumber=1577681.

[32] Christopher Y. Metz. *IP Switching: Protocols and Architectures.* The McGraw-Hill Companies, Inc., USA, 1999. 464 p. ISBN 0-07-041953-1.

[33] Monteiro Edmundo Pereira Antonio. Admission control in intserv to diffserv mapping. *International conference on Networking and Services Publication.* 2006 . [cited 01 October 2014]. Available at:http://ieeexplore.ieee.org.libproxy.aalto.fi/stamp/stamp.jsp?tp=&arnumber=1690554.

[34] Minei Ina , Lucek Julian. *MPLS-Enabled Applications : Emerging Developments and New Technologies (3rd Edition).* John Wiley and Sons, Incorporated, 2010. 628 p. ISBN 978-0-470-66545-9.

[35] Coriant. *8000 INM system description.* 2014. SystemDescription.

[36] Tellabs. Tellabs 8000 intelligent network manager. *data sheet.* 2012. [cited 27 March 2013]. Available at:http://www.tellabs.com/products/8000/tlab8000inm.pdf.

[37] Coriant. 8000 intelligent network manager. *Solution Brief.* 2014. [cited 1 September 2014]. Available at:http://coriant.com/products/documents/SB_8000_INM_SolutionBrief_74C0018.pdf.

[38] Tellabs. Tellabs 8600 smart routers. *Product Overview.*, Vol. E. 11/12:. 2012. [cited 25 March 2013]. Available at:http://www.tellabs.com/products/8000/tlab8600sysoverview.pdf.

[39] Coriant. 8600 smart router series. *Product Overview.* 2014. [cited 1 September 2014]. Available at:http://coriant.com/products/8600.asp.

[40] Coriant. 8660 smart router. *Datasheet.* 2014. [cited 29 August 2014]. Available at:http://coriant.com/products/documents/DS_8660_Smart_Router_74C0026.pdf.

[41] Coriant. 8609 smart router. *Datasheet.* 2014. [cited 29 August 2014]. Available at:http://coriant.com/products/documents/DS_8609_Smart_Router_74C0022.pdf.

[42] Intel Corporation. Packet Processing on Intel Architecture. 2014. [cited 15 Oct 2014]. Available at:http://www.intel.com/content/www/us/en/intelligent-systems/intel-technology/packet-processing-is-enhanced-with-software-from-intel-dpdk.html.

[43] Tootoonchian Amin, Gorbunov Sergey , Ganjali Yashar , Casado Martin, Sherwood Rob . On Controller Performance in Software-Defined Networks. *Hot-ICE'12 Proceedings of the 2nd USENIX conference.* 2012. [cited 15 Oct 2014]. Available at:https://www.usenix.org/system/files/conference/hot-ice12/hotice12-final33_0.pdf.

[44] Pushpinder Kaur Chouhan Barbara Fraser David Lake Jim Finnegan Niel Viljoen Marc Miller Navneet Rao Sakir Sezer, Sandra Scott-Hayward. Are we ready for sdn - implementation challenges for software-defined networks. *Whitepaper.* 2013. [cited 10 Sep 2014]. Available at:http://netronome.com/wp-content/uploads/2014/07/IEEEComms_Are-we-ready-for-SDN-Whitepaper.pdf.

[45] Evans John William, Filsfils, Clarence. *Deploying IP and MPLS QoS for Multiservice Networks : Theory and Practice.* Morgan Kaufmann, 2007. 456 p. ISBN 9780123705495.

[46] D. Katz,D. Ward. Bidirectional Forwarding Detection (BFD). *Internet Engineering Task Force: RFC 5880.* June 2010. Also available at:http://tools.ietf.org/html/rfc5880.

[47] B. Niven-Jenkins,D. Brungard, M. Betts, N. Sprecher, S. Ueno . Requirements of an MPLS Transport Profile. *Internet Engineering Task Force: RFC 5654.* September 2009. [cited 29 Aug 2014]. Also available at:http://tools.ietf.org/html/rfc5654.

[48] Soliman Mourad, Nandy Biswajit, Lambadaris Ioannis, Ashwood-Smith Peter . Exploring Source Routed Forwarding in SDNBased WANs . *IEEE International Conference on Communications (ICC)*. 2014. [cited 29 Oct 2014]. Available at:http://ieeexplore.ieee.org.libproxy.aalto.fi/stamp/stamp.jsp?tp=&arnumber=6883792.